DOS 5.0 Beta Test Program

Review and Recommendations for the Future

Microsoft Confidential

Revision: 0.2
Date: 7/29/91
Custodian: Andy Hill

CONFIDENTIAL

EXECUTIVE SUMMARY

The DOS 5.0 beta test was the largest beta program in Microsoft's history. It contained over 7500 sites covering 10 countries. The program used a number of new methods for gathering feedback from the beta sites, including the use of a Compuserve Forum to support beta sites and the practice of having the beta sites return data disks that provided machine readable data on the tester's machine and configuration. Response to the beta test, from beta testers as well as from within Microsoft has been very positive. The software was tested on an extremely broad range of machines and operating environments. Feedback from beta testers was gathered quickly and efficiently. Beta sites received competent and timely technical support over a number of communications channels. The beta test also had the added benefit of generating support and momentum for the product in both corporate accounts as well as among influential "power users".

Some of the key success factors in the beta program included:

Adequate staffing, including a full time database administrator/beta coordinator.
Use of a Compuserve forum for feedback and support
Use of a networked relational database to store and track all information on each beta tester
Use of a mandatory installation report to increase and track the level of beta tester participation
Use of telephone call-downs to obtain survey information and prompt testers to install and test the software
Requirement that testers return a diskette containing detailed information on their hardware and configuration.
Presentations to User Groups to get feedback and acquire additional, influential beta testers

OVERVIEW

A number of lessons can be learned from operating a program of this size. The following document attempts to highlight the good and the bad aspects of the program and serve as a road map and point of reference for future beta tests.

The objective of the beta program was to test DOS 5.0 on as broad a spectrum of machines and user environments as possible. The beta test was executed with the goals of receiving feedback from our testers as fast as possible and providing a high level of technical and administrative support to these beta sites. Feedback included bug reports as well as end-user response to new features and functionality of the operating system. Finally, the beta test had the secondary objective of creating support and momentum for the product as it neared commercial release.

The review is broken into 3 parts. The first part contains an overview of what we did right, what we did wrong and what we would recommend for future beta programs.

The second part gives a detailed description of the beta program, how it was organized and details on the various parts of the program.

The third section provides a summary of some of the data gathered from the program that might have relevance for other beta programs. This includes a breakdown of the types of beta testers, the type of equipment used by these testers and statistics on their response and participation.

In order to explain and evaluate the beta program, it is helpful to divide it into three phases: Acquisition, Distribution and Support/Feedback, defined below:

Acquisition
This consisted of locating and signing up beta sites, getting the proper Non-Disclosure document, logging them in a database and communicating to the sites or requestor the status of the request.

Distribution
This consisted of fulfilling the "orders" generated via the acquisition stage. This included pulling together the various components of the beta kit, manufacturing it and shipping these kits to the beta testers.

MS-PCA 2559141

Support/Feedback
This consisted of leading technical and administrative support to the beta sites and gathering feedback from these sites. Feedback was then disseminated to development, testing, user education, PSS, product management and program management.

Section I: RECOMMENDATIONS

What We Did Right: Acquisition

User Groups. We went to 9 of the largest User Groups in the U.S. and made presentations to the top 100 of the group's members. We put everyone under non-disclosure, demonstrated the product, went through Q&A and then signed everyone in attendance up for the beta test. This had a number of positive outcomes:

- It allowed us to sign up users that had very non-standard configurations and older 8088 and 80286 machines. (many corporate and ISV sites had standard configurations and only 386/486 machines.)
- It provided us with very good feedback, for the present and future versions of the product.
- It was a win-win situation: The user groups were able to give their top members a peek at an unreleased product and get them signed up as beta testers. It gave MS a captive audience of power users and a leveraged method of acquiring many beta sites at once.
- It provided great PR with the power users and heavy influencers. We feel that it also created "ownership" of the product among the testers, and will go a long way in promoting the product once it is released.

ISV Relations Group. We went to the ISV relations group and made sure that we had all of the ISVs that they considered important on the beta program.

PR - VIP List. We worked with PR and the Waggener Group to develop a VIP list. This list contained all of the critical authors and members of the press that we needed to handle carefully. All members on the beta team had this list posted next to their phones. Whenever any contact was made with a VIP, a summary of the conversation or contact was emailed to an alias that included PR, marketing, program management and the business unit vice president. The VIP list was updated weekly.

Black List. We kept a black list of all companies and individuals that should not be on the beta program. The list included certain competitors, known counterfeiters, people that had violated their NDA, etc. The black list was updated weekly and posted at each team member's desk.

BETA500 Alias. We used a special alias and email account for all internal beta site requests. Program management was on the alias and thus reviewed all new beta site requests. The email account was used to manage and track fulfillment and status of the requests. This configuration also prevented any individual in the group from having their name attached to the beta program. This prevented random email and phone calls.

Centralized Relational Database and Administrator. We used a networked relational database to store all information on the beta sites, and had a full time database administrator. The database administrator was responsible for designing the database at the onset, adding needed forms and capabilities throughout the program and ensuring integrity and standardization of all data that went into the database.

The importance of this cannot be understated. Each beta site had a unique number and all subsequent data was keyed to this number. We attempted to link all data about a site, from name and address to actual technical support requests to the database.

Categorized Sites by Type (ISV, IHV, Corp. User, Press, etc.) Each beta site was categorized in the database by its type. This was particularly valuable when we needed to make follow-up mailings to specific groups and to check relative participation of each group.

MS-PCA 2559142

CONFIDENTIAL

Weekly Reports. The database administrator generated weekly reports from the database. It listed the number of new requests, number signed up, and number pending. It also gave a listing of all beta sites and other information on participation and status of the entire beta program.

Shrink-Wrap NDA. We used a "break-the-seal" non-disclosure agreement for all non-OEM beta sites. This greatly simplified and streamlined acquiring and signing-up beta sites since a signed NDA wasn't needed. However, the legal department felt that they had limited recourse when using a license like this, and there were numerous reports of bootleg copies of beta builds in use by individuals and on bulletin boards.

Internal Beta: Trained Helpdesk, Tracked Active Sites. Before we launched the first corporate-wide internal beta test, we gave a presentation to the helpdesk managers and engineers and made sure they had disks and documentation. We also tracked all active internal beta sites, providing us with statistics of the number of testers as well as a means to contact all internal testers if we uncovered serious problems. We also used these internal testers for smaller, non-corporate-wide beta tests.

What We Did Right: Distribution

Used PRS for Betas & Mass Mailings. Product Release Services (PRS) is an incredibly flexible resource with dedicated people. They were able to build and ship over 5000 beta kits in under one week. They were also excellent for handling the logistics of other mailings to beta sites (duplication, envelope stuffing, mailing).

Used Compuserve Forum for Release Candidate Builds, Fixes, Drivers. We used the private forum on Compuserve to distribute fixes, device drivers and even entire builds (as we approached RTM). This was fast and efficient, requiring little administrative overhead.

Beta Tester Guide (Overview, Expectations, Contact Points). This 20 page document was one of the keys to the success of the program. It provided a one-stop point of information on how the beta program worked, what was expected of the beta tester and what they should expect from Microsoft. It also listed all contact points for technical support and general questions (i.e. bad disk replacement, NDA issues, etc.).

The guide was also a place where we set expectations with the user on the type of response they would receive. For example, as we moved into our final beta, we made it very clear that we were close to shipping a final product and that we only had the time and resources to collect bug reports from our testers. We stressed that we would not be able to get back to them to let them know if the bug was fixed and that we wouldn't be able to offer detailed end-user support.

"Flash Sheet" in Each Kit. On the top of each beta kit we put a "flash sheet" that contained phone numbers and points of contact for the beta testers. It was printed on a different colored piece of paper and everything was in large type. We operated under the assumption that this would be the only thing that the beta tester would read before grabbing the disks and immediately installing the software. We also used it for alerting the user to specific issues that we considered critical (i.e. read the README files on disk 1, use compuserve if you can, etc.)

Dual Media and Readonly Disks. By shipping 3.5" and 5.25" disks we made sure that the sites could get up and running asap and cut down on the overhead of tracking and shipping different disk sizes. It also had the side benefit that if a site had a corrupted disk, many times they could use the other size disks to install or get a copy of the corrupted file(s).

Readonly disks were used to eliminate the threat of virus transmission and to prevent disks from accidentally being overwritten or corrupted by the user.

Built "Diskless" Kits Towards the End. As we approached RTM, we started making kits without any disks in them. When we needed to ship a beta kit, we would insert the latest build in the kit so that we were not beta testing stale software.

MS-PCA 2559143

What We Did Right: Support/Feedback

Service Log Database. We set up a database that recorded every installation report, phone call, FAX or compuserve bug report. It logged the person's name, categorized the nature of the interaction (bug report, suggestion, user error), categorized the area in which the problem occurred and tracked the status of the resolution of the problem. It allowed us to:

- Track most active testers.
- Track response and resolution.
- Provide feedback to development that doesn't show up in the bug database (areas of confusion, frequency that a problem or bug was reported, etc.).
- Provide feedback to PSS so they can anticipate problem areas and prepare accordingly for to support the retail product.

Installation Report. This was a 4-5 page document that the tester had to fill out for each machine and return to us. It asked for machine type, memory, peripherals, etc. It then had the tester report any problems that they encountered during installation.

We found that the Installation Report was instrumental in getting timely feedback and in getting actual problems reported. We believe that many of the problems that showed up in the installation report would have otherwise gone unreported.

We made it very clear to the testers that they must return their installation report or else they would not receive future beta builds or a free shrink-wrap retail product.

Uninstall Disk. In to the installation report, we also required that our beta sites return the "uninstall disk" that was created during the MS-DOS 5.0 installation. It contained a wealth of information that was used to both diagnose problems at the specific beta site and gather statistics for the entire beta program (i.e. CPU type, memory configuration, hard disk size and partitions, etc.)

We were also able to query the database of uninstall disk information to locate machines with specific characteristics and then send these beta sites special builds when regressing bug fixes or testing high-risk changes.

The database generated from over 2300 uninstall disks has already been used by the Windows 3.1 and NT groups to get a better understanding of the types of configurations that users have (i.e. the types of information stored in the autoexec.bat and config.sys files, etc.).

Compuserve Forum. We set up a private forum on Compuserve for the beta test that was restricted to beta testers only. The forum was incredibly well received by the beta testers and provided an efficient means of supporting the testers, distributing files and providing real-time feedback to development and marketing. We also used the forum as a reality check when making product changes or deciding not to fix a particular bug.

Call Downs. We performed telephone call downs after shipping the beta kits a number of times during the beta cycle. The call downs were used to 1) Generate survey data on the users response to product (good during early tests) and 2) to prompt testers to begin testing the software immediately. The survey data was very useful in the early stages as it gave us quick feedback on general response to the product and broad areas that required improvement (usability problems, areas of confusion). The call downs later in the program were used to push testers to begin testing time critical builds as we approached final release. We were able to test a release candidate build in about one week by calling the testers about two days after they received the kits and including pre-paid, pre-addressed Airborne envelopes in the beta kits so the testers could return the installation reports as quickly as possible.

Responded to Each Service Request. We made a point of responding, in some fashion, to each support request or problem report. Many times it was simply a FAX or phone call acknowledging the report and letting the user know that we would or would not be contacting them again. The result of this policy was a perception of a high level of service and in reinforcing expectations as to the type of follow-up support the user would receive.

Support Team on Site. We had the entire technical support staff in the same building as development and testing. This facilitated smooth communication between the technical support staff and testing/development and it created lasting relationships between the PSS support person with development and testing.

Dedicated Phone and FAX Lines. While it may sound insignificant, this eliminated mis-routing of calls and FAXes, provided a direct link to the beta support group and were easily disconnected when the beta was over.

Answering Machines for Phone Lines. This allowed us to prioritize our responses and prevented the technical support staff from being "interrupt driven" by phone calls.

Separate Technical and General Support Phone Lines. This concentrated the technical support team's efforts on technical issues, not on getting new disks and answering general questions.

Sent Free Retail Product to All Active Testers. This made for happy testers, gave us leverage to get sites to return their installation reports and ensured that all sites had final code on their machines after the test was over.

What We Did Wrong: Acquisition

Didn't Plan for Corporate Preview and Pre-Release Evaluation Programs. These programs provided final code to certain critical customers during the lag between the time that the product was released to manufacturing and the product was officially announced and available in retail form. These programs require administrative and technical support after the beta program is over. These should be planned at the beginning of the beta program and adequate resources made available to support them.

Poor Communications with the USSMD Organization. We had problems communicating with the field organization. This was made worse by the number of groups that we needed to work with: CAMs, SEs, Corporate personnel located in Redmond and field offices throughout the country. There were instances where the beta group contacted corporate accounts without the CAM or SE knowing about it first. Additionally, when USSMD corporate queried the filed for new beta sites, data was lost, resulting in CAMs and customers expecting beta kits that they didn't receive.

Many of these problems have been addressed for future beta programs and USSMD has created a corporate beta coordinator position.

OEM NDAs - Poor Coordination with Account Managers and OEM Shipping. There was much confusion as to who was responsible for getting OEM NDAs signed and how these were turned over to OEM shipping and OEM license administration so beta kits could be shipped to OEMs.

We finally found that it worked best to have the beta team obtain the NDA directly from the OEM and to work very closely with OEM shipping and Licensing Administration. OEM shipping and License Administration has also amended their procedures to streamline this process.

Internal Beta: Launched Company-Wide Too Soon. When we launched our first corporate-wide internal beta (via email to ms-corp) we made a number of mistakes. First, we should have tested the particular build on a number of groups first to ensure its stability. This was done in later internal releases. Second, we didn't have contingency plan in place if there was a disaster (i.e. if the program started damaging hard disks). Finally, during the first internal beta, we made no effort to track MS employees that had installed the software. Thus, when we did have a problem (some hard disks were being damaged), we had to broadcast a message over ms-corp warning users that there were problems. This obviously hurt internal support of the product and scared employees from testing the product. If we had a list of everyone that had downloaded the software off the network, we could have just sent a warning message to those specific people.

Got Behind on Internal Requests, Didn't Update Requestors. Our internal alias for new sites had a lot of traffic and we let it get behind at times. This should have received better attention and we should have had a means to automatically update requestors on the status of their request.

What We Did Wrong: Distribution

Didn't Coordinate well with OEM Shipping, Failed to Escalate Problems. We had problems getting OEM beta kits shipments approved by OEM shipping and OEM license administration. This was related to confusion concerning how NDAs were taken care of and what the requirements were to get something shipped to an OEM. Most of this has been addressed by OEM shipping and License Administration, but a single point of contact needs to be established as well as a defined escalation path.

Poor Communications with USSMD Field. We had problems getting beta kits to the correct people in the field and additional problems when CAMs wanted to hand deliver beta kits. This was primarily due to poor data from USSMD and poor communication with the field. Most of these problems have been addressed for future beta programs.

Didn't Get SEs Beta Kits Before Their Accounts. During the first beta some corporate accounts were shipped beta kits before their SEs received the kits, resulting in unwanted surprises for the SEs and an appearance of poor corporate coordination to the customer.

SEs, CAMs Handed Out Kits Without Notifying Us. We had a situation where the field had some extra beta kits and began handing them out without notifying the beta group. This created serious problems when these testers called for support or access to the Compuserve forum and, having no record of this site, the beta team had to delay support to these sites until it was verified that they received the beta kit legitimately.

What We Did Wrong: Support/Feedback

Poor Communications with Online Administrators (accounts expired in middle of Beta Test). The Online accounts were set up with automatic expiration dates. The original expiration date was when we originally planned to ship the product. When the schedule slipped, we didn't communicate this to the Online administrators and we actually had all of the Online accounts automatically cancelled before the beta test was over. If Online is used in the future, all accounts should be set up with a 2-year license and then terminated once the product ships.

Compuserve Forum: Slow To Get it Operational, Slow to Escalate Problems. We had a number of problems working with Compuserve. These included poor response in getting the initial forum running, a scroll rate that was too fast (comments would only be in the forum for a few days before scrolling off) and severe security problems during the beta test. (At one point, all security in the forum was accidentally removed by Compuserve, allowing access to anyone. At that point we had beta builds available for download, which made the problem even more severe.)

In any future dealings with Compuserve, there should be 1) a well defined escalation path and 2) a procedure put in place that outlines the steps that will be taken to ensure security any time a change is made to the forum. This needs to be put in writing and needs to clearly define the testing procedures that will be taken to insure security.

Uninstall Disks: Poor planning on processing (had to re-process disks three times): While the uninstall disks contained a wealth of information, processing each disk took a good deal of time (each disk had to be manually inserted into a disk drive and a program run to extract the information). On the first pass, we only extracted a portion of the available data on each disk, not realizing that we might want to use the other information later. We ended up needing additional information later and had to re-process the disk two more times. We also neglected to record the size of the uninstall disks (3.5" or 5.25"), which we needed when sending the free retail copy to our beta testers.

What We Should Do in the Future: Acquisition

Ensure that Beta Site List is Complete. We discovered a number of areas that weren't covered at the beginning of the beta program:

- Meinel Top Sellers
- Certain Microsoft Competitors
- Non-MS Environments (Novell Networks, Diskless Workstations)
- OEMs
- Press
- Non-Profit (Universities, other educational sites)

External Sources of Sites: Hard Rules (Data & Deadlines). When obtaining new sites from an external source (OEM Sales, USSMD), there needs to be very well defined rules for data format and content as well as hard deadlines. This should include written instructions detailing the conventions used in entering data (Capitalization rules, allowable characters, etc.) and a sample Excel spreadsheet (or something similar) that demonstrates the required format. Additionally, requests for beta kits after the initial shipment of beta kits should be expected and planned for accordingly (a buffer stock of beta kits needs to be kept on hand to fulfill these requests.)

Liaisons From USSMD, SEs, OEM Sales & Mktg: Aliases. At the beginning of the beta program there should be a single person from each of these organizations established as the primary point of contact. These people should be placed on all of the status aliases used in the beta program and should act as the speaker for their respective constituents. These people need to be involved in planning the beta program and should be kept abreast of any schedule changes, etc.

CPP, Pre-Release, Evaluation Programs Planned Ahead of Time. These programs need to be part of the overall beta/development plan from the beginning. Resources and ownership of these programs need to be assigned early on, including technical support and administrative overhead.

International Sites. International sites should be planned and acquired at the onset. Coordination with the subsidiaries will be necessary and the logistics of shipping and support need to be addressed.

Request Alias: Better Tracking, Notification. An Eform should be made available for Microsoft people to submit requests. Some form of automatic email notification should be used to notify the requestor that the request has been received, what will happen next, and a time frame for this to occur. Once a site has been accepted/signed up or denied, the original requestor should be notified.

What We Should Do in the Future: Distribution

Ship Updated Builds with New Kits. There is a point in each beta cycle and as you approach RTM that it is no longer useful to ship "old" builds and you will have to start shipping recent builds with each new beta kit. The tradeoff is the support problem of having bugs reported against numerous versions, and having other beta testers hear that a new build is available and wanting it for themselves.

PRS: Treat Like Kings: T-Shirts, Mugs, etc. The people at PRS are great and should be considered a valuable asset. Make sure that they get T-shirts, Mugs, etc. and that they are taken out to lunch, etc. after the beta program is completed.

Work with SSBU Verification and OEM Shipping. These two groups need to be included in beta program planning from the beginning and included on all status aliases.

Internal Beta: Create Log of all People that Download from the Network. There should be an automated method of logging the email name of every person that downloads beta software from the network (i.e. \\toolsvr\beta). This will create statistics for the degree of internal testing, provide a list of people that can be used in smaller tests and make it easier to communicate with these people if serious problems in software show up.

What We Should Do In the Future: Support/Feedback

Compuserve Forum from the Beginning. We didn't begin using a Compuserve forum until the third and final beta. It should be used from the onset and should be positioned as the preferred method of reporting bugs and obtaining support.

Installation Reports: Machine Readable and Serialized. One complaint that we had from our beta testers was that for each successive beta build that we sent out, we required them to fill out a new installation report each time, meaning that they had to report the same information about their machine configuration repeatedly. In the future we should make the first installation report machine readable (on disk possibly) and then serialize each test machine so that the user will be only have to fill out a short "update installation" for each successive shipment.

CD ROM Storage for Uninstall Disk Information. When stripping the data from uninstall disks (or other information gathering devices), the disks should be read only once and all info stored to disk (or CD ROM depending on the volume of info). We ended up having to process the uninstall disks 3 times because we failed to capture all of the data on the first pass.

Track Activity of Beta Testers and Remove Inactive Testers From Program. While we shipped beta kits to a large number of sites, less than half provided the minimum feedback by returning their installation reports. We threatened to stop sending future builds to testers that didn't return these, but didn't follow through on these threats (however, we did not send free retail product to sites that failed to return an installation report). Considering the cost and overhead of shipping a beta kit, and the need for as many active beta sites as possible, it may make sense in future beta programs to drop inactive testers during the beta test and replace them with new sites.

Call Downs. Call downs to all beta sites should be scheduled for one week after each shipment. This should encourage beta sites to install the software and begin testing, and speed the entire test cycle. Additional phone surveys can be used later in each cycle to get feedback on the product. These really shorten the time it takes to get feedback from testers and also assures them that someone at Microsoft is really interested in their input.

Get Software/Hardware Immediately. One recurring problem was obtaining hardware and software needed to reproduce bugs reported by beta sites. The beta team needs to be pro-active in getting this as soon as a problem appears that might require the hardware or software on-site. In a few instances we needed to have the beta testers ship us their machines. In these situations, a process needs to be set up so that these testers are reimbursed for shipping and for the rental of a replacement machine while we use theirs for testing. All shipments should be insured.

Better Use of Service Log. While the service logging database was extremely useful, we could have made even better use of it by 1) generating standard weekly reports that would be distributed to development, testing and marketing and 2) linking this database to the master database. We had technical problems with this, but it shouldn't be hard to implement in future beta programs.

PSS Should Supply Technical Support Team. Out of the five technical support specialists, only one was a full time PSS employee. The other four were temps that we had to hire and train on our own. In future beta programs, PSS should provide all of these people. At the very least, they should be responsible for recruiting, interviewing and hiring the temps. There should be an incentive for PSS to do this: it allows them to train people on the next version of the product and also allows their full time people to develop personal relationships with their counterparts in development and testing.

Use FAX Cards. Use of FAX cards for both incoming and outgoing data would speed response time and cut the amount of paper floating through the project. While FAX boards present a security problem for MIS, a solution that allows their use should be found.

Fast Databases. Much of the day-to-day administrative and technical support tasks depend on fast access and processing by the centralized and service log databases. The faster the database, the more efficient the entire group will be.

Prevent Beta Testers from Going Directly to ISV/IHV's. We had a number of cases where a beta site would discover a problem with a particular piece of hardware of software and would contact the vendor directly. This needs to be discouraged and clearly stated in the beta tester guide. It needs to be made clear that we need to make sure that any problems are reported to the beta support team and that we will work with other hardware and software vendors to resolve problems.

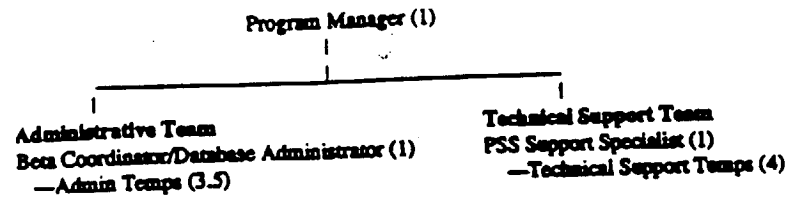
Tracking Mechanisms in Bug Database. When entering a bug in the bug database, the source of the bug needs to be logged (i.e. internal beta vs. external beta) and a contact person, with phone number, should be included in the report.

MS-PCA 2559149

SECTION II: DESCRIPTION OF THE PROGRAM

Organization

The Program had the following organization:



The program was broken into two basic teams: Administrative and Technical. The two teams ran fairly independently of each other, with an emphasis on utilizing the technical support team for dealing with technical issues and minimizing their administrative work.

The Administrative Support team was responsible for processing the various outgoing and incoming non-technical information as well as processing the returned installation reports and uninstall disks.

The Technical Support team was responsible for providing technical support to the beta sites, gathering feedback from these sites, summarizing this data and forwarding it to development, testing, program management, marketing and product support.

Using the model of acquisition, distribution and support/feedback to describe the beta program, details of each component are discussed below.

ACQUISITION

We acquired our beta sites from a number of channels, including:

- BETA500 alias for internal requests
- incoming phone and mail from external requests
- outgoing phone and mailings to acquire specific key accounts
- Participants in previous Microsoft beta tests
- Corporate sales force (USSMD) for corporate accounts
- OEM sales force for key OEMs
- ISV Relations Group for key ISV
- User Groups via presentations to select members
- Education Account managers for colleges and universities
- Public Relations and Corporate Communications for Press and Authors

As new sites were acquired, they were entered into the central database.

Non-disclosure Agreement (NDA) Process

During the first beta test, a traditional NDA that required a signature was used. During the second and third beta, we employed a "break-the-seal" NDA for all non-OEM beta testers. This was a one page license agreement that was included in the beta kit. There was an activation sticker on the outside of the box that prompted the user to read the enclosed licensing agreement.

The use of the break-the-seal NDA greatly streamlined the process of signing up new beta sites, eliminating the time and paperwork involved in obtaining a signed NDA.

Database
We used a centralized relational database throughout the beta program. It operated over the network, allowing access to all members of the beta program. The core of this database was the initial information on a company that was entered during the acquisition stage and the ongoing information on shipments and status on each site. Section III has a screen dump of the general information recorded for each site.

It was an extremely time-consuming process to design the database, enter data and maintain the data throughout the beta program. However, the capability to generate reports and have all data centralized and available to the support teams was tremendously valuable.

DISTRIBUTION

External
We used Product Release Services (PRS) to build and ship the external beta kits.

In general, the beta kits contained the following:

- "Flash Sheet," listing phone numbers and support channels.
- Break-the-Seal Non-Disclosure Agreement
- Cover Letter
- Packing list with a phone number to call if any items were missing
- Beta Test Guide
- Installation Report
- Beta Forum Compuserve Users Guide and pamphlet w/ trial membership account
- Airborne Express Envelope and pre-addressed airbill (for returning the installation report and disk)
- Beta Software: Dual media (3.5" and 5.25") READONLY disks
- Beta documentation

A detailed description of the contents follows.

Flash Sheet. We assumed that the user would only read this sheet. This was a bright colored, single sheet of paper that was the first thing that the beta tester saw when opening the beta kit. It listed the phone and FAX numbers to call for both technical and non-technical support and was used to point out any other information that we felt the beta testers must see. This really helped channel all support calls to the proper lines and gave the testers a single point of reference for reporting problems.

Break -the-Seal Non-Disclosure. This was a one page licensed mentioned earlier. We also put a sticker on the outside of the beta kits across one of the seams, warning the user that by breaking the sticker, they accepted the terms of the enclosed licensing agreement.

Cover Letter. This basically outlined the contents of the beta kit, reiterated the fact that everything should be considered confidential and highlighted the most important aspects of the program (i.e. that they should return the installation report immediately).

The cover letter should not be signed by an individual. We made the mistake of signing someone's name to the cover letter that went out with the first beta, and that person's phone rang for two months. We signed all subsequent letters "MS-DOS 5.0 Beta Test Group."

Packing List. This was a checklist of all items in the beta kit. It contained a phone number to call if they were missing anything. The phone was connected to an answering machine.

Beta Test Guide. This 20 page document was key to the success of the program. It outlined:

- New features in the product
- The benefits of being a beta tester
- The responsibilities of being a beta tester and what is expected.
- Microsoft's responsibilities and the type of response to expect from the beta team.
- Installation instructions
- Problem reporting instructions and phone/FAX/Compuserve numbers to call

The test guide was a one-stop shopping place to get questions answered, reducing the number of calls and inquiries for the beta support team. It also allowed us to set expectations at the onset as to what level of support the test sites will receive. In the early betas, we were able to give more support, but as we moved into the final beta, we made it very clear that we were strictly in a bug-gathering mode, and that testers should not expect us to get back to them with resolution on a particular bug that they reported.

Installation Report. This was a 4-5 page document that the tester had to fill out for each machine and return to us. It asked for machine type, memory, peripherals, etc. It then had the tester report any problems that they encountered during installation.

We found that the Installation Report was instrumental in getting timely feedback and in getting actual problems reported. When performing follow-up with beta testers that didn't have to fill out an installation report, we discovered that many of them encountered problems during installation or with early use of the product and abandoned it, without ever reporting the problems. With the installation report, we could follow up with sites that were having problems and get them running and testing the software. We feel that many bugs and problems would have otherwise gone unreported if we had not used the Installation Report.

In addition to the installation report, we also required that our beta sites return the "uninstall disk" that was created during the MS-DOS 5.0 installation. The uninstall disk had valuable information needed when diagnosing a problem, in addition to general information on the particular machine. We were able to record CPU type, amount of memory, hard disk size and partitions, etc. This information was extremely valuable to testing/development. It allowed us to verify the types of machines that were being tested and also gave us the ability to identify particular machine configurations that would be affected by changes. In one situation where we had found a bug that only affected certain obscure machine configurations, we were able to query the database of uninstall information, identify the specific users and send them an updated build to make sure that their systems worked with the new changes.

Beta Forum Compuserve Users Guide and pamphlet w/ trial membership account. The Users Guide gave general instructions for accessing the beta forum. The pamphlet was provided by Compuserve and included detailed instructions for new users on how to log in and set up an account. Included in each pamphlet was a temporary account and password with a $15 usage credit (although the actual beta forum was free, beta testers were charged the prevailing connect rates for any time spent on Compuserve outside of the beta forum.) The temporary account numbers were all individualized and pre-coded by Compuserve so when a user logged-on with one of these numbers, they were granted automatic access to the beta forum. (If a "normal" compuserve user tried to access the beta forum, they were forced to fill out an application form that was then reviewed by the support team. Only legitimate beta testers were then given access to the forum.)

Airborne Express Envelope and pre-addressed airbill. This was included to make it as fast and easy as possible for the beta site to return the Installation Report and Uninstall Disk. The Airbill was pre-addressed to our beta test group and was pre-paid.

Beta Software: Dual media (3.5" and 5.25") READONLY disks. We shipped everything on dual media to cut down on the administrative overhead of identifying the disk size that each site required and to eliminate the need to re-ship different sized disks (again cutting down on administrative work and allowing the beta site to install the software as soon as they received it). This also had an added side benefit: if a beta site received a bad disk, they could often recover the corrupted files or perform the installation from the other set of disks. Again, this cut our work and got the sites testing as soon as possible.

MS-PCA 2559152

CONFIDENTIAL

We used readonly disks to eliminate the possibility of virus problems. It also prevented the user from accidentally erasing files or overwriting the distribution disks.

Beta documentation. We shipped 8.5x11" photocopies of the beta documentation and asked beta sites to report any problems found in the documentation in the same manner that they reported problems with the software.

Internal Distribution
We announced the availability of the beta software to Microsoft employees via corporate-wide email (ms-corp alias). The software was put on the \\toolsvr\beta share as well as a share owned by the development/testing group.

Due to the large distribution of the third external beta, one of the requirements before we shipped it was a successful internal beta test. This internal beta test consisted of installing the beta software on a number of groups, making sure that there weren't any problems with the software before going to a wider audience. Only after this was successful was ms-corp email sent announcing the corporate-wide beta test.

When groups were upgraded, a developer, tester or program manager would assist the tester in getting the software installed if needed. After installation, if anyone encountered a problem that couldn't be resolved quickly over email or phone, someone from the development group would go to that person's office to get their machine operating or to get information on the bug.

SUPPORT/FEEDBACK

The support/feedback portion of the beta used a team of technical support people that monitored and responded to beta site input and support needs over a number of channels.

The Team
The team consisted of 5 support specialists. One was a full time person on loan from PSS and the remaining four were temps. The temps were first screened by PSS for technical ability before hiring.

Channels
Information and communication with the beta sites came over a number of different channels. We attempted to steer as many people to Compuserve forum for support. This was done because the Compuserve forum allowed us to provide support on a priority, "non-interrupt driven" basis. The nature of the forum also encourages other beta testers to provide assistance and support to their fellow testers.

Compuserve Forum.
Overall, the Compuserve forum was one of the most successful aspects of the beta program. It offered a fast, efficient means of communication with our beta testers and provided development, marketing and program management with immediate, real world feedback.

Over 1,000 users joined the Compuserve forum, sending over 12,000 messages (4,000+ threads).

Additionally, as with other compuserve forums, the support burden was significantly reduced because other testers ended up answering and supporting their fellow testers. We also had program managers and developers log into the forum and answer/clarify questions from our testers.

The forum was also very useful in getting a "reality check" when we were making changes to the product or deciding to not fix certain bugs. We could broadcast messages mentioning that we were considering adding or deleting a feature or not planning on fixing a particular problem. Depending on the responses to these broadcasts, we could alter our plans if needed.

Furthermore, the feedback from the forum acted as a valuable motivator for developers, who were able to see immediate, non-filtered feedback on their work.

The forum was also used to make fixes, additional drivers and even new builds available for download and quick testing.

We set up a private forum on Compuserve dedicated to the DOS 5.0 beta test. The forum had a number of features that "regular" public forums didn't have:

Free. We negotiated a per-user, per-month rate with Compuserve and made the forum free to the beta testers. This was done to lower barriers to usage and encourage extensive participation in the forum.

Security. The forum was set up with a number of security measures to ensure that only legitimate beta testers gained access. We set up two methods of accessing the forum. For beta testers that didn't have an existing compuserve account, they could use the compuserve intro pamphlet enclosed in each beta kit. These pamphlets were provided by Compuserve and each one had an unique account number and password that allowed the new user to dial in to compuserve and set up an account. All of these accounts were pre-configured to gain access to the beta forum.

For beta testers with existing compuserve accounts, there was a gateway at the entrance of the forum that checked to see if the person attempting entry had a pre-approved account (from the intro pamphlets mentioned above) or if their account number had be added to a membership database maintained by the beta technical support team. If their account didn't satisfy either of these requirements, they were sent to a survey that prompted the user for their name, company, address, phone, etc. After filling out the online application, a message was given that we would review their application and admit them to the forum after Microsoft approval. If the applicant was a legitimate beta tester, their account would be added to the membership database and they would receive unobstructed entry into the beta forum.

Online Problem Report Form. An online problem report form was developed with Compuserve that asked beta testers for pertinent data about their systems and a description of the problem they were experiencing.

In actual operation, this online problem report form had little use, and mostly by those not familiar with compuserve. The problem report form was a one-line-at-a-time questionnaire that was long and difficult to change answers once they had been entered. Compuserve is limited in this respect in that they are restricted to a TTY type user interface. We also made a copy of the questionnaire available for downloading. Users could fill out the form offline, and then upload the completed form when done.

The problem report form was also not an optimal solution because it did not make the problem "public" for other forum members to see, comment on and even propose a solution. For these reasons, we gravitated towards receiving most of the problem/bug reports in actual forum messages.

Phone
We had a dedicated line with an answering machine attached. This allowed us to prioritize responses to callers and prevented the support team from being "interrupt driven" by phone calls. When the beta test was completed, the message on the answering machine was changed to notify callers that the product had been announced, the beta program was completed,and that they should contact PSS for additional assistance.

FAX
We used a dedicated phone line and FAX machine. Initially we went through the general Microsoft FAX line (883-8101) but had recurring problems due to misroutings by telex and incomplete routing information provided by the beta site on the FAX. As with the phone, when the beta was over, we disconnected the FAX line, permitting a clean shut down of the beta program.

Online
We made the option of a free Online account for the duration of the beta program available to our initial 700 beta sites and all OEMs . Few sites took us up on the offer (about 50 out of the initial 700 sites, with a total of about 100 by the end of the program). PSS supported the service requests that came over Online and escalated any unresolved SRs to the beta technical support team.

DOS5BETA Alias (for internal beta test)
For internal bug reporting and technical support of the internal beta test, we used an alias that was used as a filter before a bug was actually entered into the bug database. The entire beta technical support team was on this alias, as were the development and test leads as well as the entire program management team. It was policy within the group that an internal complaint or problem received immediate response.

Managing The Data

Incoming calls, FAXes, Installation reports, etc. were handled by the technical support staff and all incoming information was logged in a service database. This database allowed us to:

- Track all requests for resolution and proper service
- Identify non-bug problems and frequent areas of user error and areas of confusion.
- Generate timely reports on the status of our beta testers
- Identify our most active beta testers
- Feedback for PSS product launch

Whenever an installation report, service request, bug report, suggestion, etc. was received from an external site, it was recorded in the service logging database. Information captured in this database is detailed in Section III. Captured information included name, phone number, type of contact (bug, suggestion, user error), the area in which the problem occurred (installation, networks, shell, etc) and a notes field that gave a brief description of the problem. If an installation report was received, the relative degree of success was recorded as well.

We also used the service log database to track resolution of the service requests, with each entry tagged as open or resolved. This helped in making sure that sites received the proper response and that service commitments didn't slip through the cracks.

This service log database provided valuable, timely feedback to development, testing and program management. While any bugs were immediately logged into the bug database, the service log captured all problems, including user error, areas of confusion and volume of problems for each particular area.

It also allowed us to track our most active beta sites, which were then selected as *gamma* testers. In the *gamma* tests we selected our 500 most active sites for a final, quick turn around test of a release candidate build.

Entering Problems in the Bug Database
The distinction between the bug database and the service log database needs to be made clear. The service log database was used only by the beta team to track and log all service requests. The bug database, on the other hand, was the RAID-based database used by testing and development to track actual changes to the software. Not all problems reported by a beta site were entered into the bug database, only those that appeared to be actual bugs in the software, as opposed to user error, etc.

If the technical support team had a suspected bug, they would search the bug database for a duplicate of the same problem. If no duplicate existed (i.e. if this was a new bug), the technical support team would enter the problem into the bug database.

Processing Installation Reports and Uninstall Disks
Upon receipt of an installation report and uninstall disk, the administrative team logged each response in the master database and sent a copy immediately to the technical support staff (this was done to make sure that the beta site receive immediate attention if they had encountered problems during installation and it also allowed any problems to be reported to testing/development as soon as possible.) The uninstall disk was then processed by a program that stripped the needed data from it and entered it into a large database. The installation report and uninstall disks were then numbered, cross referenced with the master database and filed for later use if needed. (In many instances, if the user was having installation problems, we could use the uninstall disk to diagnose/reproduce the problem here at MS).

MS-PCA 2559155

Staffing

During the final beta test, we had the following breakdown managing the support channels:

Compuserve	2 people
Phone	1
FAX	1
Internal Beta, escalated Online issues	1

MS-PCA 2559156

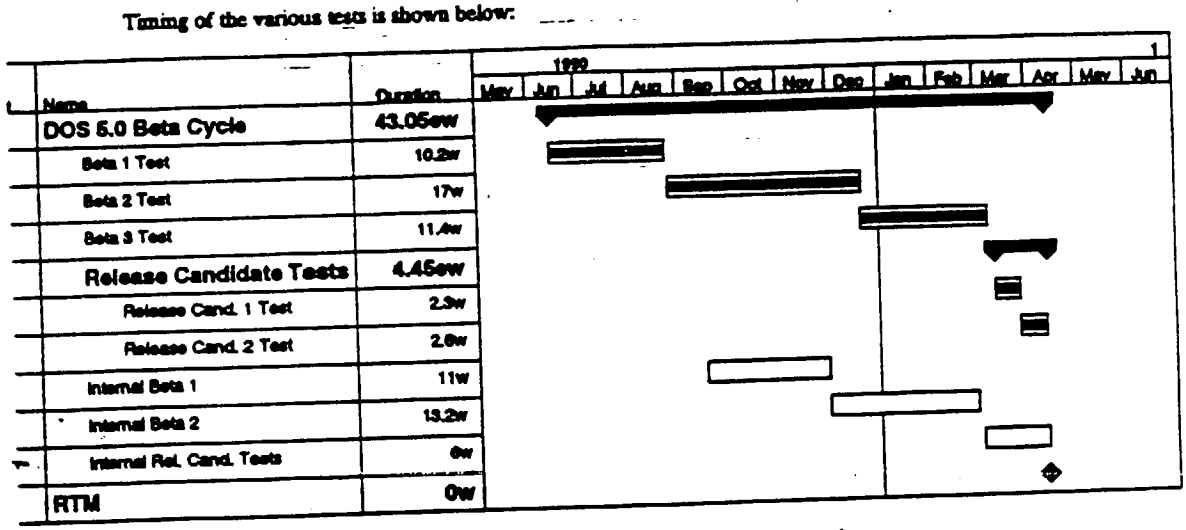
Section III: SUMMARY DATA

Releases and Beta Tester Profile

There were three major external beta releases and two major corporate-wide releases. The third beta included final functionality. In addition to the three beta tests, a final *gamma* test was distributed to the most active 500 beta sites and to the ISVs/IHVs with the top selling products just prior to Final Test Entry. The following table shows how many testers tested each version:

Beta Test	Shipped	Number of Testers
Beta 1	Jun '90 - Aug '90	700
Beta 2	Sep '90 - Oct '90	1400
Beta 3	Dec '91 - Feb '91	7600 (30% international)
Gamma	Mar '91	500 (20% international)

Timing of the various tests is shown below:



The following table shows the distribution of beta testers, categorized by type:[1]

Description	Number of Testers
Corporate sites	2100
Beta test kits distributed internationally by Microsoft subsidiaries.[2]	2300
Individual software vendors (ISVs)	1100
Members of user groups	900
Original equipment manufacturers (OEMs)	500
Individual hardware vendors (IHVs)	250
Non-profit organizations including colleges and universities.	250
Book authors and magazine editors	200
TOTAL:	7600

1 Numbers are rounded off to the nearest 50. Numbers reflect number of beta testers. Some sites have more than one beta tester.

2 Beta kits were distributed in the following countries: the United Kingdom (350), France (375), Germany (375), Australia (175), Canada (175), Netherlands (125), Sweden (175), Italy (250), Spain (125), Mexico (125), and Brazil (100) [numbers rounded to nearest 25]. Within the countries, distribution among corporate, OEM, ISV/IHV, etc. was similar to distribution in US.

Equipment Used by Beta Testers

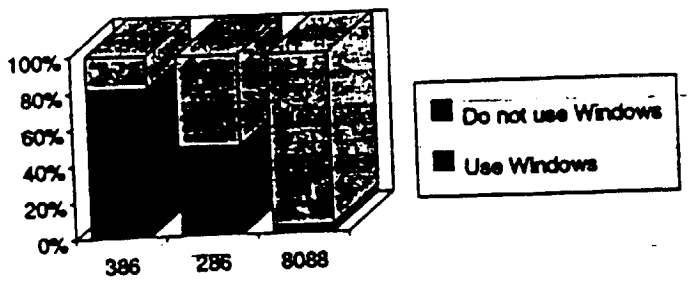
The systems in use ranged from vintage IBM PCs (some of them upgraded to XTs) to 33Mhz 486s. ISA, EISA, and MCA based hardware was used. Displays ranged from text only monochrome through XGA, Super VGA, and higher resolution,. Fixed disks ranged from 5 Megabyte to over 1 Gigabyte.

The chart below shows how the hardware was distributed (by CPU) among the beta testers:

The high percentage of 386 (which includes 386SX/486) platforms is consistent with the industry movement to the 386 platform. In addition, it is appropriate for an operating system that will be distributed mainly for use on 386 (and better) platforms.

Approximately 80% of the beta testers who were using 386 or better machines were also using Windows. In fact, of the initial 700 beta testers, 300 were selected specifically because of their participation in the Windows 3.0 beta test. The following graph shows the percentage of beta testers who used any version of Microsoft Windows, broken out by the CPU of their hardware:

Percentage of Beta Testers Using Windows, by CPU



Beta Tester Response Level and Activity

Beta tester response and activity was measured by two criteria: installation report returns and problem reports/service requests. The following table lists these measures for the third beta:

BETA 3	Total Sites	Installation Reports Returned	Problem Reports/Service Requests
Domestic	5300	2300 (43%)	1100 (21%)[3]
International	2300	1100 (48%)	n/a

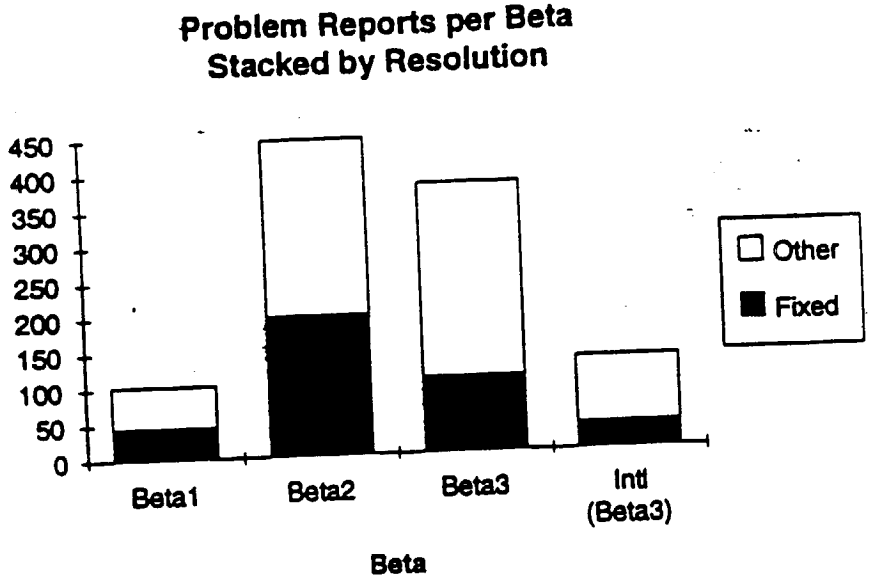
The following chart shows the number of problem reports which came from beta sites that were actually entered into the bug database, stacked by the final resolution of the problems[4]: These were problem reports that weren't initially determined to be user error. The reduced number of problems in the later beta reflects how the product stabilized over time. The drop in the fixed problems is especially significant when considering that the third beta went to over five times as many testers as the second beta. The low number of problems reports which resulted in fixes to the code reflects user confusion (often a result of the pre-release documentation and other issues with installing MS-DOS on a system which is already in use). In some cases, there were problems with third party software and/or problems in hardware.

---

3 This figure represents the number of beta testers that reported a problem or requested service. The total number of problem reports and service requests was higher due to a single site reporting multiple problems.

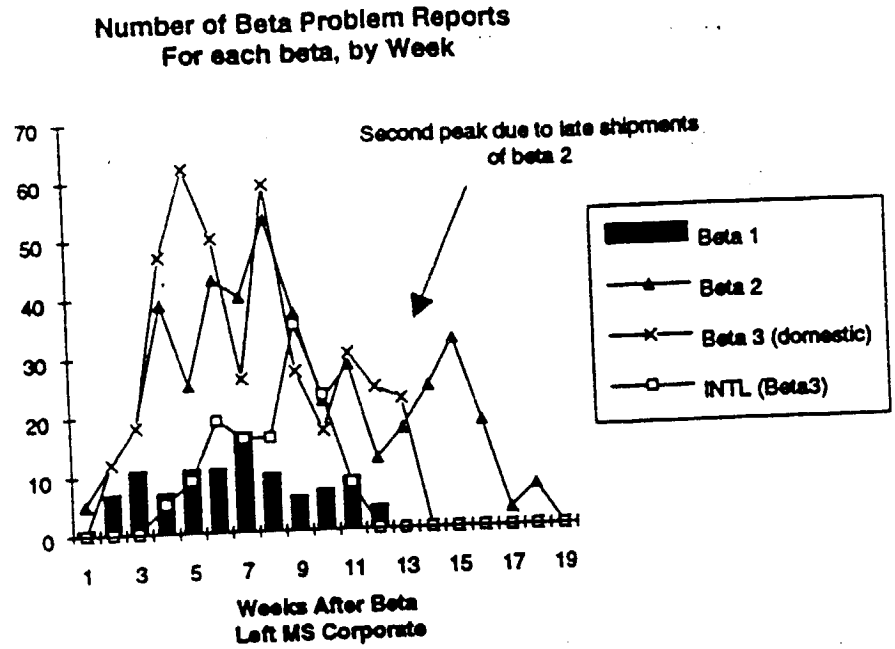
4 Problem resolution "Other" means problems which were user errors, duplicate problem reports, and problems which were not reproducible.

Many of the reported problems were reproducible on earlier versions of MS-DOS and were eventually determined to be problems with the application software and/or hardware rather than with MS-DOS 5.0. In other cases, since MS-DOS 5.0 utilized the machine more than previous software, it ran into hardware problems that had never appeared before.



Reporting Cycle

Beta problem reports were tracked from the beginning with Beta 1. The rate of problem reports for each beta peaked six weeks after shipment of the beta.[5] The following chart plots distribution of problem reports for the major betas, by the number of weeks from shipping the original beta to receiving the problem reports. The height of the bars/lines reflects the total number of problem reports received that



week.

The beta testers used MS-DOS on thousands of different configurations. Many of the problems were configuration specific. The most common configuration problems involved A20 handling (necessary to run DOS the high memory area), networks, video adapters, 3rd party disk drives, older device drivers and software, programs which do not work correctly if loaded below 64K, and applications that never anticipated a DOS version greater than 3.x or 4.x.

---

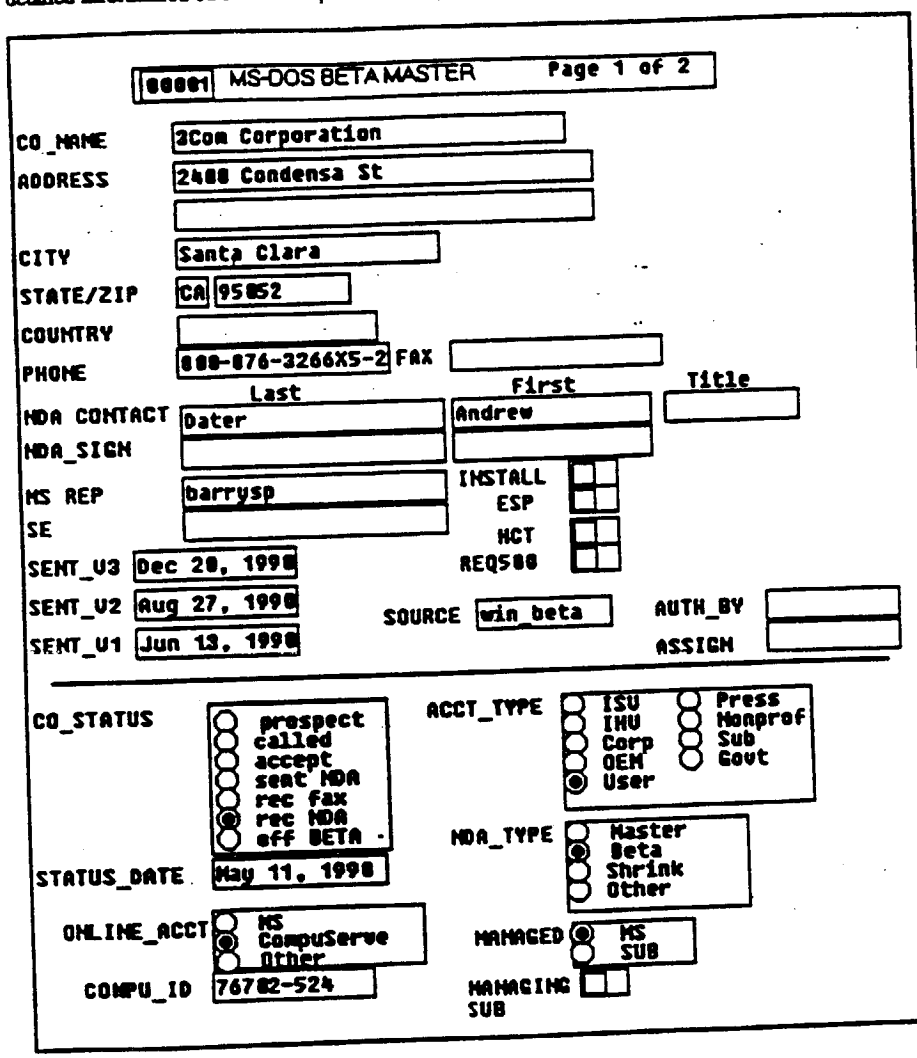
5 There is some variance due to ongoing shipments of beta kits to new beta sites after the initial beta shipment date; The peak for international problem reports was slightly later.

Database Information

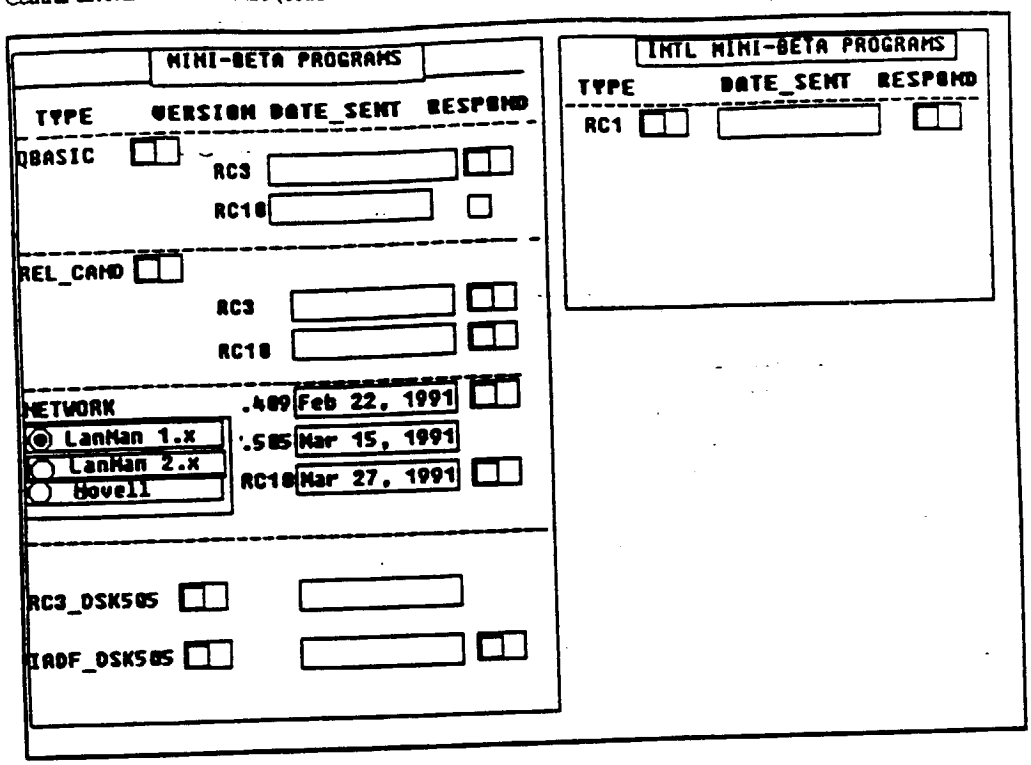
The following sections show the data and layout of the information captured for the central database and the service log database. It should be noted that the service log database was separate from the bug database (RAID) used by development and testing.

Central Information Database

The following screen shots show the general information captured in the central database. Additional notes and detailed information on smaller shipments and updated builds were logged in additional forms.

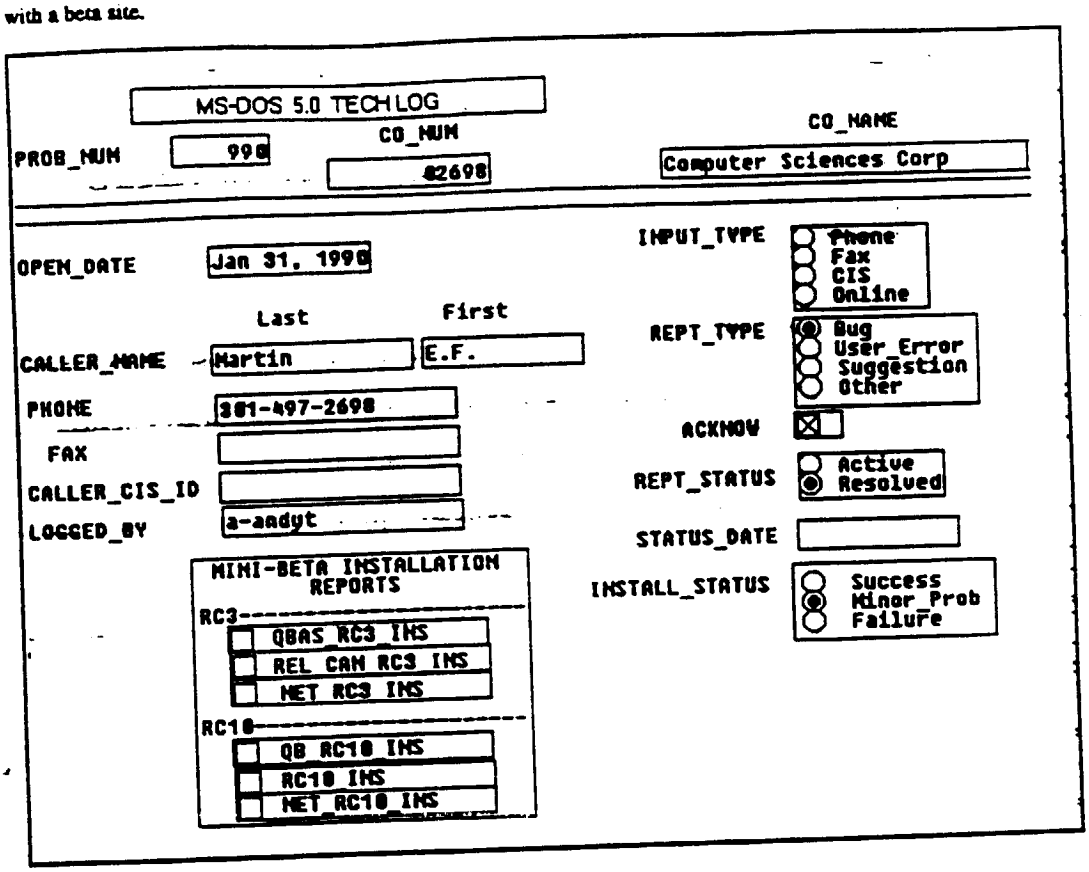


Central Information Database (continued)

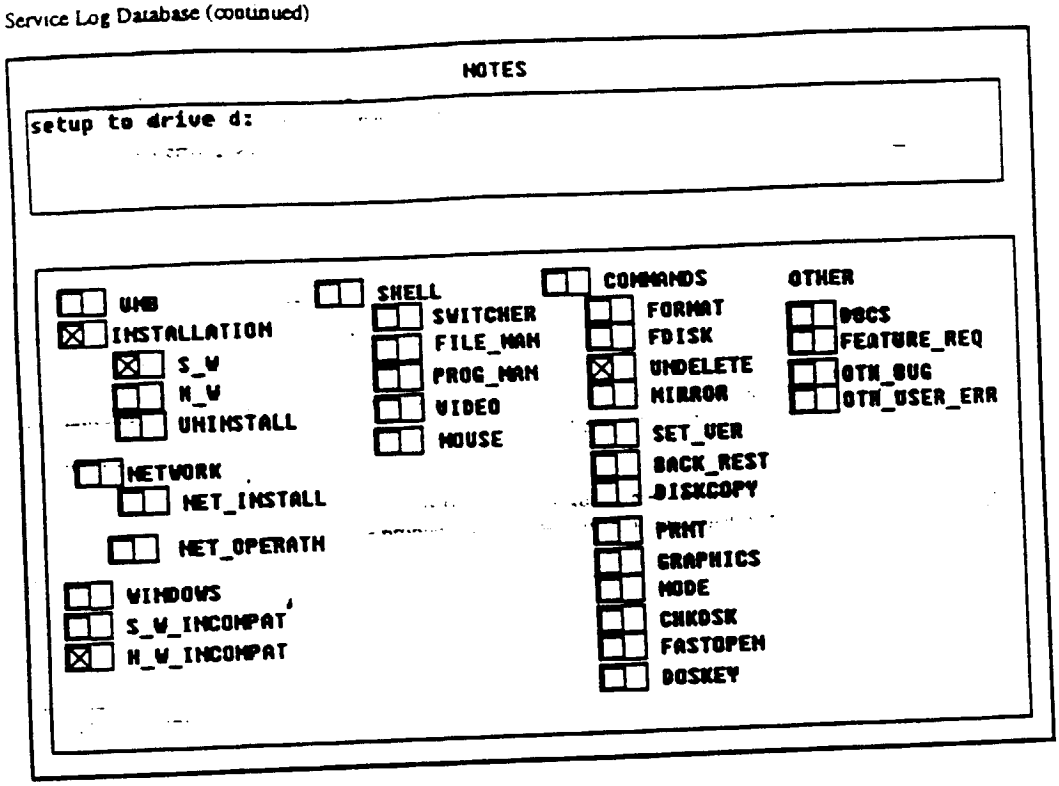


Service Log Database

The following screen shots show the data gathered whenever the beta technical support team had any interaction with a beta site.



Service Log Database (continued)



CHECKLIST FOR BETA SHIPPING - DOS/WINDOWS

TASK	SUBTASK						BEFORE SHIPPING		WHO	DATE DUE	DONE
Prepare list of documents							60 days				
Prepare Beta Kit materials											
Assign docs to staff							60 days				
Tracking Docs											
	Q & A										
	Getting Started/Beta Tester Guide										
	Installation Report/Beta Test Report Form										
	Bug Form										
	Letter										
	Flash Sheet										
	Return Mailer										
	MSD/Test Disk										
	NDA Shrink and label										
	Questionnaire										
	Release notes										
Pre-Shipping Activities General											
	Warn PRS, PRM, OEM Shipping,SMSD that Beta shipping						30 days				
	Give PRS/OEM Shipping specifics of beta						21 days				
		names of shipments									
		approximate quantity									
		contents of kit									
		# of disks/type media									
		shipping date									
	Work out duplication schedule of major docs and disks with PRS						14 days				
	Send major docs to PRS to be duplicated						14 days				
	Arrange calldowns (internal w/temp reps or external w/Telemarketing)						30 days				
	Arrange for mailers and Word Processing to do merge letters/labels/co number labels						14 days				
Pre-Shipping Activities Database							7 days				
	Dump all sites										
	Freeze database and add no more sites										
	Change message on alias line to say new sites will be added when shipping over										
	Dump hard copy of sites and addresses and contacts										
	Screen hard copy for duplicates and cut duplicates										

CHECKLIST FOR BETA SHIPPING - DOS/WINDOWS

TASK	SUBTASK	BEFORE SHIPPING	WHO	DATE DUE	DONE
OEM Shipping		7 days			
	Generate pre-shipping list of OEM's				
	Send list to OEM order entry				
	OEM ordering compares list against their approved list				
	Reconcile differences in lists	7 days			
Internal Test					
	Smoke disks				
	Run small test in group/building				
	Set up problem reporting channel/alias				
	Set up general question channel/alias or phone				
	Send ms_corp announce				
	Put code on network or dup on disks for testers				
	Run corporate test				
	Check bugs				
	Fix code	7 days			
Shipping activities general					
	Confirm former shipping schedule with PRS				
	contents, quantity, shipping date				
	Disks to PRS				
	All docs to PRS				
	Build overbuild count into quantity and stick to this #				
	Request enough copies for SE's, US field offices and foreign subs, plus MS internal				
	Request shipping list from beta team				
	Announce to SMSD that beta is shipping				
	Set order of shipping (which kits, which sub groups)				
	Ship first to SE's, MS field offices, and Subs				
	Arrange foreign shipments	7 days			
Arrange Calldowns - final					

CHECKLIST FOR BETA SHIPPING - DOS/WINDOWS

TASK	SUBTASK						BEFORE SHIPPING		WHO	DATE DUE	DONE
							3 days				
Shipping activities - database											
	Generate hard copy shipping list with SB query and save query										
	Break list down into small groups (~ 500) with simple queries for use to update shipping records										
		Mailing Services requests 81/2X11 copy long way									
	Generate separate lists for OEM shipping and PRB Manufacturing										
	Scan against NO PRE-RELEASE INFORMATION LIST and for accuracy										
	Update CO_SHIP file with shipping query from shipping list										
	Keep spread of shipping lists, queries, quantities and dates										
	Put shipping list on network for review by SMSD										

Windows 3.1 Final Beta Release Survey
Win 3.1

Name:_____ MS Caller:_____ Completed Survey_____

Company:_____ Date Contacted:_____ Left Message_____

Co #: _____ Msg Left (Date):_____ Not Reached

Phone:_____

Beta 2 Kit contents: Retail disks, MSD disk, Beta Test Guide, Information Sheet, Letter

On the 1st call:

If leave message, say:

Hello this is _____ with the Microsoft Corporation. I'm calling about the Windows 3.1 Final Beta Release sent to you on 12/26. If you have received the kit we sent, please install it and return the MSD disk as soon as possible. Please let us know immediately if you have any problems. We prefer you send them via Compuserve, but we will also accept them by fax. If you did not receive the kit, send us a FAX at (206) 869-8475 with your name, Company, phone number and size of diskettes required. Thank you.

Questions:

1. Have you received the Windows 3.1 Beta 2 Kit sent to you on 12/26?
   _____Yes _____No

(If yes, go to next question.)

(If no, please confirm name and address and say we will ship them another kit.)
(Terminate)

2. Do you need 3 1/2" or 5 1/4" diskettes in future windows 3.1 software?
   _____3 1/2" _____5 1/4"

3. Have you installed the Windows 3.1 software yet?
   _____Yes _____No

(If yes, go to next question.)

(If no, say the following) Please install it as soon as possible and fax us the feedback or put it on CompuServe immediately.
(Terminate)

4. Was the installation successful?
   _____Yes _____No

(If yes, go to next question.)

(If no, record comments and go to next item.)

5. Please report any feedback on this program immediately on CompuServe or fax feedback to (206) 869-8475. Thank you.

Ask testers to report problems as above. For other questions, ask testers to refer to numbers listed on Info Sheet.

(Do not give the below Info Numbers)
Technical Support: (206) 936-3440 - messages only
General Questions: (206) 936-7154 Fax: (206) 869-8475

MS-PCA 2559169

P:\BETA\DOCS\CALLDNS\WIN31_B2
FBR_SURV.DOC (Revised 12/30/9

CONFIDENTIAL

CONFIDENTIAL

Windows 3.1 Final Beta Release
PLEASE INSTALL and TEST IMMEDIATELY

This is the Final Beta Release and needs to be installed and tested immediately so we can find all remaining problems before shipping the final retail product.

- ***Obtain Your Company ID Number From the Mailing Label on This Beta Kit.***
  This is on the bottom right of the mailing label and is marked "CO # = XXXXX".
- ***Read the Enclosed Beta Test Guide.***
  This contains detailed information on installation, problem reporting and being a beta tester. Details on all of the points below are in the Guide.
- ***Test Any Problems That You Have Previously Reported.***
  Re-submit complete problem reports if they still exist.
- ***There Will be Limited Technical Support for this Beta.***
  We have 12,000 beta testers; you should expect delays in response and limited assistance.
- ***Use CompuServe to Report Problems and Get Assistance.***
  It will receive priority in response and processing.
- ***Use REPORT.EXE to Format all Problem Reports.***
  This is contained on the Documentation Disk. Use it for CompuServe, FAX and Mail.
- ***Return the Final MSD Disk Within 2 Weeks.***
  This will qualify you for a free retail copy of Windows 3.1.
- ***Use the Phone Only in an Emergency.***
  The only time that you should call the technical support line is in an emergency where you cannot reboot your machine or face a loss of data.
- ***Release Notes are contained on the Disks.***

SETUP.TXT	On the Windows 3.1 Disk 1
README.WRI	"Read Me" icon in Main Group
NETWORKS.WRI	In your windows subdirectory

- ***Updated Documentation Provided in Write (.WRI) Files.***
  Follow instructions on the Documentation Disk label.
- ***Additional TrueType Fonts Disk Included.***
  See the LUCIDA.TXT file on the *Additional TrueType Disk* for more information.

Contact Information:	
To Access the CompuServe Windows 3.1 Beta Test Area, type	! GO WINBETA
To Submit Problem Reports via FAX:	206-869-8475 (FAX)
Non-Technical & General Questions:	206-869-8475 (FAX) 206-936-7154 (answering machine)
Emergency Technical Support:	206-936-3440 (answering machine: for emergencies only)
Address	Microsoft Corporation, Attn: WINDOWS 3.1 BETA TEST GROUP 3/1 One Microsoft Way, Redmond, WA 98052-6399

MS-PCA 2559170

CONFIDENTIAL

This is a legal agreement between you (either an individual or an entity), the end user, and Microsoft Corporation. If you do not agree to the terms of this Agreement, promptly return the disks and the accompanying items (including written materials and binders or other containers) to Microsoft Corporation.

MICROSOFT LICENSE AGREEMENT
(FOR PRE-RELEASE Windows 3.1)

1. GRANT OF LICENSE. Microsoft grants you the non-transferrable right to make up to ten (10) copies of the enclosed program (the "SOFTWARE") solely for use at one location for evaluation purposes only. The SOFTWARE shall not be copied or used for any other purposes, and Company agrees to destroy or erase all copies of the SOFTWARE upon written request by Microsoft. You may not rent, lease, sell, sublicense, assign, or otherwise transfer this SOFTWARE or accompanying written materials ("Documentation"). You may not reverse engineer, decompile, or disassemble the SOFTWARE.

2. PRE-RELEASE CODE. This SOFTWARE is pre-release code and is not at the level of performance and compatibility of the final, generally available product offering. The SOFTWARE may not operate correctly and may be substantially modified prior to first commercial shipment.

3. COPYRIGHT. The SOFTWARE is owned by Microsoft or its suppliers and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g. a book or musical recording) except that you may copy the SOFTWARE as provided in this agreement.

4. CONFIDENTIALITY. The SOFTWARE and Documentation are confidential and proprietary to Microsoft. You agree not to disclose or provide the SOFTWARE, Documentation, or any information relating to the SOFTWARE (including the existence of the SOFTWARE or the results of use or testing) to any other party without Microsoft's written permission. Unauthorized use or disclosure of the SOFTWARE or Documentation may cause irreparable harm to Microsoft. You agree to promptly report any unauthorized use or disclosure to Microsoft.

DISCLAIMER OF WARRANTY

Because the SOFTWARE constitutes pre-release code and may be changed substantially before the first commercial release, the SOFTWARE and Documentation are provided "AS IS" WITHOUT WARRANTY OF ANY KIND. MICROSOFT FURTHER DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE SOFTWARE AND DOCUMENTATION REMAINS WITH YOU.

IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE OR DOCUMENTATION, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

U.S. GOVERNMENT RESTRICTED RIGHTS

The SOFTWARE and Documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(i) and (2) of Commercial Computer Software – Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Microsoft Corporation/ One Microsoft Way/Redmond, WA 98052-6399.

This Agreement is governed by the laws of the State of Washington.

Should you have any questions concerning this Agreement, or if you desire to contact Microsoft for any reason, please write: Microsoft Corporation Windows 3.1 Beta Test Group/One Microsoft Way/ Redmond, WA 98052-6399.

10/29/91

MS-PCA 2559171 CONFIDENTIAL

February 11, 1992

Dear Beta Tester:

Enclosed is the Final Beta Release of Windows 3.1. All features have been implemented, our goal now is to find and fix any remaining problems. Thanks to your help in finding bugs in previous beta releases, we've been able to deliver what we think is a very stable final beta release. Please install this final beta release right away and begin testing immediately. We would like you to use this final beta release for your everyday work. If you encounter any problems, please report them immediately to the beta team. See the included *Beta Test Guide* for instructions on how to do this. Since this is the final beta release, this is your last chance to report problems.

Compatibility with applications written for Windows 3.0 is extremely important to us. Please be sure to test all of your Windows applications extensively with this beta release and let us know how well they do or don't work.

As with previous beta releases of Windows 3.1, all of the information and materials included in this package are subject to the terms of the "Microsoft Corporation Non-Disclosure Agreement" or "Microsoft License and Confidentiality Agreement for Pre-Release Windows 3.1" that you or company have accepted as a Windows 3.1 Beta Tester. Please honor these agreements.

Enclosed in this beta kit is:

- The Windows 3.1 Final Beta disks
- The Windows 3.1 *Beta Test Guide*
- A *Microsoft Diagnostics Disk* that you will need to run on your machine and return in the enclosed pre-paid mailer.
- The *Documentation Disk*, which contains updated documentation files in .WRI format.
- The *Additional TrueType Disk*, which contains some additional TrueType fonts you can use to test Windows 3.1 with.
- A membership brochure and instructions for setting up an account on CompuServe.
- The Microsoft License and Confidentiality Agreement for Pre-Release Windows 3.1.

To get started using this final beta release of Windows 3.1, please do the following:

1) Read the *Beta Test Guide*. This guide is your "manual" on how to install the final beta release, how to send problem reports via CompuServe or FAX, and how to be a great Windows 3.1 beta tester.
2) Install this beta release following the instructions in the *Beta Test Guide*.
3) Once you have installed the final beta, view the *release notes* by double-clicking the "Read Me" icon in the Main Group. The release notes contain information on new features added since the previous beta release, plus known problems and workarounds.
4) Keep the one-page *Beta Program Notes* for future reference.
5) If you find any problems, follow the instructions in the *Beta Test Guide* on how to reproduce the problem and submit a complete problem report.

Special note to Novell network users, please read the SETUP.TXT file on Disk 1 prior to installing this final beta release. This file is a standard text file and is not compressed so you can view it with most word processors.

So that you can better test TrueType in Windows 3.1, we've included some extra TrueType Fonts. See the file LUCIDA.TXT on the *Additional TrueType Disk* for information on how to install and use these fonts. We've included these additional fonts so that you can test the final beta release of Windows 3.1, but do not count on these being in the final Windows 3.1 product.

Thank you for participating in the beta program for Microsoft Windows version 3.1. With your feedback we have improved Windows 3.1 usability, application compatibility, hardware support, and reliability. We hope to repay your testing efforts with a great final Windows 3.1 product for you and your organization.

Sincerely,

Windows Beta Program

Microsoft Windows 3.1 Beta Test Guide
Final Beta Release

Microsoft Corporation

Contents

NOTE: This is beta software and, as such, has not been completely tested. Sites using this software may encounter minor problems and it is possible that some sites may encounter loss or destruction of data. You should expect some down time and should back up your system prior to installing the beta software, and often during the beta test.

1.0 Quick Instructions

This is the Windows 3.1 Final Beta Release. The product is very stable and all features that will be included in the final product are in this release. We are in the final stages of the development cycle and are at the point where we need to uncover all remaining problems and fix them for the commercial release.

We need you to thoroughly test Windows 3.1 by using it as your everyday computing environment and reporting any problems **immediately** to the Beta Support Team. Brief instructions follow:

1. Read this guide thoroughly. It will save both you and the Beta Support Team considerable time and problems if you take the time to read this and adhere to the requirements of the beta program.

2. Install the new software immediately (Section 2.0).

3. Generate MSD logs and return to Microsoft (Section 3.0).

5. Test this version by using Windows 3.1 as your everyday computing environment (Section 4.0).

6. Report any problems that you find immediately (Section 5.0).

   If you do find a problem, take the following steps:

   a) Try to reproduce it (Section 5.1)

   b) Generate a complete problem report using REPORT.EXE (Section 5.2)

   c) Submit the problem report to Microsoft (Section 5.3), using one of the following methods:

   - CompuServe (Section 5.3.1)
   - FAX (Section 5.3.2)
   - U.S. Mail (Section 5.3.3)
   - Phone (Section 5.3.4, emergencies only)

2.0 Installating the Beta Software

First, a word of caution. Although we have put much effort into the verification of this software, it is still pre-release software which will be used by most applications that you run.

Sites using this software may encounter minor problems and it is possible that some sites may encounter loss or destruction of data. You should expect some down time and should back up your system prior to installing the beta software, and often during the beta test. Microsoft is not responsible for any problems resulting from the use of this beta software.

To install the Windows 3.1 beta software, do the following:

1. Back up any critical data/programs prior to running the installation (better safe than sorry).
2. Run CHKDSK/F from the MS-DOS prompt (without Windows running) to make sure that your hard disk does not have any cross-linked files. Correct these if any are found or see your MS-DOS Documentation for more information.
3. Read or print out the SETUP.TXT file on disk 1 (it describes how to make certain applications work, known problems, etc.)
4. Insert disk 1 into your a: drive.
5. Type **a:** and press the enter key.
6. Type **setup** and press the enter key.
7. Follow the instructions issued by the installation program. You can get online help by pressing the F1 key.

   Please test Setup by:

   a. Installing over your existing Window 3.0 directory. We recommend backing up your drive, group files (*.GRP) and initialization files (*.INI) before doing this.

   b. Installing into a new directory.

Using the Documentation

The beta documentation for Windows 3.1 is included on one of the beta disks and is in Windows Write (.WRI) format. You can print or view these files using the Windows Write Word Processor.

These files are distributed on the disks in compressed format; you will have to run a short installation utility to expand and copy them to your hard disk.

To install the documentation disks to your hard disk:

1. Make sure that you have an existing directory for the documentation files. If you don't, create an new directory (i.e. type "MD C:\DOCS")
2. Insert the Documentation Disk in your floppy drive.
3. Move to your floppy drive (i.e. type "a:")
4. Type **INSTALL <path>**, where path is an existing directory where you want to have the files copied (i.e. type "INSTALL C:\DOCS")

The files will be expanded and copied in to C:\DOCS.

See the file BOOKS.WRI first for a complete list of the files provided and the corresponding chapter titles.

If you find errors or problems with the documentation, please report them in the same manner that you would report operating problems.

3.0 Returning the MSD Disk

We require that each beta site run the Microsoft System Diagnostic utility (MSD) on each machine that they will be testing the software on and submit the output to Microsoft. The output from MSD provides vital information about what configurations the beta software has been tested with, the configurations that have been missed, and helps us identify memory-resident software that should be added to our testing suite.

The log file created in the procedure below captures information about your hardware, network, CONFIG.SYS and AUTOEXEC.BAT files. It does not record any other type of information. Please feel free to look at the log file before sending the disk to Microsoft.

To run MSD and create a log file for return to Microsoft, follow these steps:

1. Insert the MSD Information disk into your floppy drive.
2. Change to floppy drive (i.e. type a:).
3. Type MSD
4. Choose Print Report by typing R,
5. Choose Generate to File,
6. Enter file name MSD.LOG.
7. You will be asked to give your name, company name and company number.
8. MSD will record the your computer's configuration information to the MSD disk.
9. Write your Company Number and Company Name on the disk label (be sure to use a felt-tipped pen).
10. Mail this disk back to Microsoft in the envelope provided. If the envelope is not available, you can mail the disk(s) to:

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
Attn: Windows Beta Program 3/1

Please do this for every machine on which you run Windows 3.1. You will need to use a unique file name for each machine, i.e. MSD1.LOG, MSD2.LOG, MSD3.LOG, etc.

4.0 Testing the Beta Software

Once you have installed this version of Windows 3.1, we ask that you test it by using it as your everyday computing environment.

Compatibility with Windows 3.0 Applications

One of the primary goals for the Final Beta Release is to test compatibility with Windows 3.0 applications. Our goal is for ALL Windows 3.0 applications to run perfectly on Windows 3.1 and we need your help in finding any compatibility problems that might exist.

Please try to run all of your existing Windows 3.0 applications under this Beta release, and let Microsoft know if you encounter any problems. It's important to really use the application: use it to do the kinds of things you do day-to-day such as creating new documents, loading and editing existing documents, printing, etc.

Problems Reported Against Earlier Beta Versions

If you have previously reported a problem when using an earlier beta version, please test to make sure that those problems have been fixed in this version.

If a problem that you have previously reported still exists, please re-submit a new problem report describing the bug.

5.0 What to do if you find a problem

If you encounter any problems using this version of Windows 3.1, we want to find out about it immediately. However, in order for us to fix the problem, we need you to submit a complete problem report describing the problem, the configuration and environment that the problem occurred under and the steps needed to reproduce the problem (if possible).

If you encounter a problem that you cannot reliably reproduce, or that only occurred once, you should still submit a problem report containing the exact configuration that the problem occurred under.

The following sections have detailed instructions on how to reproduce a problem, create a problem report in the proper format and submit the report to the Microsoft Beta Support Team.

Note: ***If you encounter a problem or conflict with Windows 3.1 and hardware or software from another vendor, you may not contact that vendor to discuss the problems.***

You should report the problem in the manner described in this guide. Microsoft will contact the third party (and is probably already working on the problem with the third party.) We require this so that no problems go unreported to Microsoft. It is also a violation of the Non-Disclosure Agreement to discuss the beta program or software with anyone else.

5.1 Reproducing the Problem

Please follow the steps below to simplify the configuration that you are using to attempt to narrow down any problem you might find. Doing this will really help the Beta Support Team and Microsoft development in diagnosing and fixing any problems you uncover.

Simplify the Configuration

When reporting a problem (bug), try to reproduce the problem with the simplest system configuration. Try removing all unnecessary drivers and TSRs from your config.sys/autoexec.bat files, except the himem.sys driver in your config.sys file. Remove subst, smartdrv, ramdrive, append, share (if you are using MS-DOS 5.0), etc. Remove any "stacks=" statements from your config.sys.

Stripping most things out of your config.sys and autoexec.bat isn't the easiest thing to do, but this is a way of possibly tracking down the cause of the problem.

One way to simplify this process is to keep your minimal config.sys and autoexec.bat files on a bootable floppy disk and boot from that disk when checking your system against problems you've found.

The ideal autoexec.bat and config.sys files to use when testing against a problem are:

CONFIG.SYS:

```
files=30
buffers=20
device=c:\win\himem.sys
REM Use the himem.sys from your Windows 3.1
REM  directory
```

AUTOEXEC.BAT:

```
prompt $P$G
path <your path>
```

Of course, there will be exceptions to this; you may have a printing problem on a network printer where you must have your network drivers loaded; you may have a drive controller which requires that you load a special driver in your config.sys in order for your computer to interface with one or more of your disk drives. You shouldn't remove these drivers when testing.

Using this method, you might be able to come across the source of your problem.

Try to Find the Cause by Putting Things Back into autoexec.bat and config.sys

If you do strip most commands from your config.sys and autoexec.bat files and the problem goes away, you should start adding the deleted items back, one at a time, until you (hopefully) find the source(s) of the problem.

Please be sure and mention if altering your config.sys and/or autoexec.bat did anything to change the behavior of the problem.

Experiment with win.ini and system.ini

Advanced Users may also try experimenting with their win.ini and system.ini files in trying to isolate and/or "fix" problems. If you are able to fix a problem in this manner, please submit a full problem report anyway. We need to be aware of these types of problems.

Obtain a Dr. Watson Log (if Present)

If you have a reproducible problem, and it is producing Dr. Watson output, follow these steps:

1. Delete the Drwatson.log file from your windows directory.
2. Run through the steps necessary to duplicate the problem. You should now have a small (2-10Kb) DRWATSON.LOG file, containing information specifically about your problem.
3. Include this file when you report the problem to Microsoft.

See Appendix B for more information on the Dr. Watson utility.

Report the Problem to Microsoft

Follow the instructions in the following section to create a problem report using REPORT.EXE and send your problem to the Windows 3.1 beta team.

NOTE: ***CompuServe is highly recommended and will receive priority processing.***

5.2 Generating a Complete Problem Report using REPORT.EXE

The beta support team has developed a simple, MS-DOS based program that compiles problem report and system information files that you create along with your vital system files (AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, SYSTEM.INI, DRWATSON.LOG) into a single report that you can then upload into the forum.

Use of this is program is highly recommended; it will save you time in compiling all of the files and information required by the Beta Support Team and will provide your information in a consistent format for processing at Microsoft.

Obtaining REPORT.EXE

A copy of REPORT.EXE has been included on the Documentation Disk. If you have installed the documentation files, REPORT.EXE will be in that directory. Otherwise, you can just copy REPORT.EXE from the Documentation Disk to your hard disk. REPORT.EXE, can also be downloaded from the Windows 3.1 beta forum. It is stored as REPORT.ZIP (a compressed format).

Using REPORT.EXE to Process Problem Reports

1. Create a small text file that contains your system configuration. A text file is a file created by using Windows Notepad, or your favorite text editor. Please DO NOT use a Write formatted file, or a file formatted by any other type of word processor. Use only a straight text (ASCII) file.

   The System Configuration File should contain:

   - The version of MS-DOS that you are using (ex: MS-DOS 5.0).
   - The build of Windows 3.1 you're using (you can get this from the Program Manager Help.About menu item).
   - Your machine name/type (ex: Genesis 386/40).
   - Your BIOS brand/version/date (ex: Phoenix BIOS v 1.1.2 dated 3/6/88).

     NOTE: The BIOS brand, version, and date is often displayed when the computer first starts. You may also be able to obtain this information by running the MSD utility. Also note that out of brand, version, and date, brand and date are the most important.

   - The type of Hard drive and controller (ex: 124mb Maxtor HD with WD 1007 controller).
   - The type of Video Card/Monitor and the mode that your machine is running (ex: Trident 8900 card with Emerson VGA in 640x480 mode)
   - The type(s) of floppy drive(s) (ex: 1.44 A: and 1.2 B:).

     Note: Floppy information is usually not important, but if your problem is specifically with your floppies, it would help to have as much information on your floppy drive(s) as possible.

- The brand/type of Mouse and the mouse driver version (if any) (ex: Microsoft Bus Mouse using version 8.00 driver). Be sure to mention which COM port it's on if it's a Serial Mouse.
- Please mention any other peripherals (net cards, etc.) that are on your machine.

A sample System Configuration File would look like:

MS-DOS 5.0
Windows Build 3.10.061
Genesis 386/40 with AMI BIOS dated 3/91
Maxtor 124 MB HD with IDE controller
Trident 8900 card with 1 MB and Emerson VGA in 640x480 mode
1.2 mb A: 1.44mb B:
Microsoft Serial Mouse on COM2 with 8.00 driver

2. After you have created the System Configuration File, you will need to create additional text files for problem reports. A problem report file should include the following:

a. A description of your bug.
b. The steps needed to reproduce the bug. Please be sure to give the exact steps necessary to reproduce the bug.
c. Whether the bug is reproducible ALWAYS, SOMETIMES or NEVER. This is a critical piece of information required to properly track down the problem.
d. A description of all 3rd-Party applications, etc. involved. Please include:

- Name and version number of the application you were running.
- The name of the company that manufactured the application.
- If special data files are required, let us know how we can obtain a copy. If possible, upload a copy of the data files required.
- If you are running on a network, the network brand and version numbers for the drivers you are using.
- If a Shareware application is involved, please include the application, either by zipping it in with the report or uploading it separately with the completed report.

If you have several problems to report on an identical configuration (config.sys, autoexec.bat, hardware, etc. all matching), you may put all of them into a single problem report. If your report contains multiple problems, each should be detailed as if it were the only problem in the report.

3. Once you have a text file for the given bug report, simply save it into your REPORT.EXE directory, and type at the MS-DOS prompt:

```
REPORT <REPORT NAME> <CONFIGURATION NAME> <WIN 3.1 DIR> <BOOT DIR>
```

where:

<REPORT NAME> is the name of the text file that describes the bug.
Example: BUG.TXT

<CONFIGURATION NAME> is the path\name of your configuration file.
Example: SYS.TXT

<WIN 3.1 DIR> is the path of your Windows 3.1 installation.
Example: C:\WINDOWS

<BOOT DIR> is the path\boot directory
Example: C:\

If you don't include this parameter, the default is C:\

The first three arguments are mandatory.

Advanced users can cut this a step further by writing a batch file or macro that contains:

```
@echo off
report %1 sys.txt c:\windows
```

where sys.txt is the name of your configuration file, and c:\windows is the directory containing Windows 3.1.

For example, if you created the above batch file called REP.BAT, you would give the command "REP BUG.TXT", where BUG.TXT is the text file containing your problem report.

IMPORTANT:

- Run REPORT.EXE using the correct System Configuration File
- Please run REPORT.EXE with the simplest config.sys/autoexec.bat that you could reproduce the problem with.

4. The newly generated problem report takes the name CHOPIN.TXT, unless a CHOPIN.TXT file already exists, in which case it will prompt you for a new one. This is the file that you will need to upload to CompuServe or print out and FAX or mail to Microsoft.

   If the problem involves a Shareware application, or specific data files needed to reproduce the problem, be sure to include copies with the report.

5. If submitted via CompuServe, check your CompuServe Mail at least once a week after reporting a given bug or set of bugs, as Microsoft may be want to contact you for more information.

6. Other notes on REPORT.EXE
   - Your personal information is saved in a file called "info.txt", and will be generated as a result of several prompts the first time you run this program.

- Having the fourth command-line argument <BOOT DIR> will allow you to choose a directory other than C:\ in case you've booted from a drive other than C:\. This is helpful if your have booted from a floppy disk with your minimal config.sys and autoexec.bat files on it.

5.3 Submitting Problem Reports to Microsoft

NOTE: ***You should expect minimal technical support during the final beta phase. We are equipped only to receive problem reports (bugs) and assist users who have critical operating problems or face a loss of data.***

Our focus at this stage in the development cycle is on uncovering and fixing problems (bugs) for the commercial release. This focus and the large number of beta sites (more than 12,000) precludes Microsoft from offering traditional end-user support for Windows 3.1 beta testers. Beta test sites should not expect to receive assistance in using Windows 3.1 or in answering general questions about the software or why certain features were/were not included in Windows 3.1. While we welcome your suggestions and comments, most of the features are now set for the final release of Windows 3.1. Any suggestions that you make are recorded for future versions of Windows but will probably not make it into the commercial release of Windows 3.1.

If you have a non-technical problem or question, such as a missing or bad diskette, please leave a message at:

206-936-7154 (Non-technical and General Questions)

We will get back to you promptly. Alternately, you can fax your inquiry to:

206-869-8475 (FAX)

When contacting Microsoft or reporting any problems, please try to include your unique company ID #. This is on the bottom right of the mailing label on your beta kit and is marked "CO # = XXXXX". If you are not the primary contact, the company number can also be obtained from the person to whom the beta kit was addressed.

5.3.1 Reporting Problems Using CompuServe

Uploading complete problem reports into a library on the Windows 3.1 Beta Area on CompuServe is the preferred method of reporting problems. Please create a problem report using REPORT.EXE (details on how to do this are in a previous section) and upload it to the appropriate library in the forum.

Please do not post problems as messages in the Forum. Any problems should be reported by uploading a complete problem report to the library that most closely matches the type of problem you are experiencing.

NOTE: Please do not upload problem reports to Library 14. It is for non-reproducible DrWatson Reports only.

Appendix A contains detailed information on accessing CompuServe and the Windows 3.1 Beta Test Forum.

5.3.2 Reporting Problems Using FAX

Beta testers who do not yet have access to CompuServe should submit problem reports (bugs) via FAX. Please use REPORT.EXE to create a complete problem report, print the file created by REPORT.EXE and FAX the printed document to:

206-869-8475 (FAX)

We use character recognition software to convert FAXes to an electronic format. Please do not submit handwritten FAXes--the text cannot be easily converted which slows processing time.

FAXes may be sent to Microsoft 24 hours per day.

5.3.3 Reporting Problems Using U.S. Mail

If you do not have access to CompuServe or a FAX, you can submit your problem reports via U.S. Mail. Please use REPORT.EXE to create a complete problem report, and mail a disk or printout containing the output file from REPORT.EXE to:

Microsoft Corporation
ATTN: Windows Beta Program 3/1
One Microsoft Way
Redmond, WA 98052-6399

We use character recognition software to convert hardcopy problem reports to an electronic format. Please do not submit handwritten problem reports--the text cannot be easily converted which slows processing time.

5.3.4 Reporting Problems Using Phone (Emergencies Only)

We ask that CompuServe, FAX or U.S. Mail be used to submit problem reports. However, we have added a phone line for emergency situations where you face a loss of data or an unusable machine. This phone will be attached to an answering machine, and we will be responding to problems based on the severity of the problem. Messages may be left 24 hours per day at the following number:

206-936-3440 (Emergency Problem Reporting)

NOTE: ***Please use the phone only if you have an emergency that requires immediate attention. Examples of emergencies are a crashed machine that you cannot get to reboot or a situation where you face a loss of data.***

6.0 General Information About the Beta Test Program

6.1 Objectives of the Beta Test Program

Our objectives in performing this beta test are to:

- Evaluate effectiveness of this software and documentation.
- Test for compatibility with existing Windows 3.0 Applications
- Verify this software in many environments.
- Verify the accuracy of the user documentation.
- Maximize confidence in the basic reliability and compatibility of this software.
- Understand areas where we can improve the software.

There are several advantages for those who participate in the Windows 3.1 beta program, including:

- Access to a pre-release version of Windows 3.1.
- Corporate testers and end users have the opportunity to verify that the new Windows 3.1 works well in their environment.
- Software developers are able to verify their product(s) on the unreleased Windows 3.1 and provide feedback to Microsoft to help assure that the product(s) run well on the upcoming release of Windows 3.1.
- Hardware Manufacturers have the opportunity to verify that Windows 3.1 installs and works well on their hardware.

6.2 Responsibilities of Beta Testers

As a Windows 3.1 Beta tester, you are expected to install and test the software immediately and report any problems according to the guidelines in this document.

In addition, according to the non-disclosure agreement, we require that you:

- Ensure that the beta software is used only by the beta test participants at your site in accordance with the non-disclosure agreement that you accepted by opening and keeping the package. (Some sites have master non-disclosure agreements which cover this beta test. At these sites, anyone who is covered by the master non-disclosure agreement may use the beta software.)
- Keep all information pertaining to the beta test and results of the beta test confidential between your company and Microsoft personnel.

- Understand that the increased levels of contact (such as calls initiated by Microsoft to you) are for the beta test period only. At the conclusion of the beta test, normal product support will be available.
- Allow Microsoft personnel access to the designated beta test equipment if necessary.
- Report any problems that you find using the software with another vendor's hardware or software to Microsoft.

  You cannot contact the third party vendors directly.

- Upon notice from Microsoft, destroy all of the beta software and beta documentation.

6.3 What To Expect From Microsoft

6.3.1 Technical Support

The Windows 3.1 beta program now has over 12,000 test sites. It is important to understand what our objectives are in this final phase and what to expect in the way of support and response from Microsoft.

The primary emphasis of the final beta test is uncovering operating problems (bugs) in Windows 3.1 and compatibility problems running Windows 3.0 applications, and fixing these problems for the final commercial product. We are not staffed to provide typical end-user support on how to use the software or to answer questions pertaining to the structure or rationale behind specific product features.

Our technical support staff is structured to receive and respond to operating problems on a priority basis, which means that situations where a machine will not boot or a loss of data is threatened will receive priority over other problems. If there is a problem with Windows 3.1, we will work to fix the problem internally at Microsoft. Under normal circumstances we will not ship new versions of the beta software that will fix your specific problem, unless it is of a critical nature. The fix will appear in the final commercial release of Windows 3.1.

Due to the large number of test sites, you may encounter considerable response delays when reporting a problem.

The CompuServe Forum will contain considerable information on how to use the beta software and hints on getting around typical end-user problems.

6.3.2 Obtaining a Free Copy of Windows 3.1

We will be providing our active beta testers with the opportunity to receive a free copy of the commercial release of Windows 3.1 once it has been released to the public. However, only sites that return completed **MSD Disk(s)** within 2 weeks of receipt of the beta kit will be eligible for the free copy of Windows 3.1. Simply reporting a problem does not qualify you for a free retail copy of Windows 3.1.

You must return the completed MSD Disk(s) within 2 weeks of receipt of the beta kit in order to qualify for a free copy.

We will be shipping only one free copy of Windows 3.1 to each registered beta sites that returns the MSD disk(s), regardless of the number of machines that were used for testing the beta software at that site. That copy will be shipped to the registered contact at that site.

Appendix A: Using the CompuServe Forum

For your convenience, we have set up a special forum on CompuServe for reporting any problems/questions you may have about Windows 3.1.

> CHARGES: *You will not be charged for connect time while in the Windows 3.1 Beta Test Area. However, any time spent on CompuServe outside of the area will be billed at the prevailing CompuServe rates.*

Obtaining a CompuServe Account

Existing CompuServe Users

If you are an existing CompuServe user, you should use your existing account to access the forum.

New CompuServe Users

If you don't have a CompuServe account, you will need to obtain one. An Introductory Membership booklet has been included in your beta kit for those of you who do not have an account with CompuServe. The booklet provides detailed instructions on accessing CompuServe and a temporary account and password that includes $15.00 of free connect time. You will need a modem to access CompuServe.

Once connected to CompuServe, you will be lead through a number of electronic forms where you will provide the information needed to start an account. With an active account, you can access not only the Windows 3.1 Beta Forum, but any other available area on CompuServe. **While the time spent in the Windows 3.1 Beta Forum will be free, you will be charged for any time you spend outside the forum.**

Accessing the Windows Beta Forum

Once you have a CompuServe account, to enter the Windows 3.1 Beta Area, type at any CompuServe "!" prompt:

! GO WINBETA

And follow the instructions on the screen. If you have provided your CompuServe ID to Microsoft at the time you signed up for the beta test, you may already have been given access, otherwise you will receive instructions on how to apply for access. When you apply for access, you will be asked for your Company ID Number, your company name and the name of the addressee that received the beta kit. (If the beta kit was not addressed to you, make sure that you include the name of that person).

NOTE: After gaining access to the Windows 3.1 Beta Test Area, you can access the forum directly by typing !GO WINBTUSER. You can still access it by typing !GO WINBETA as well.

Using the Windows 3.1 Beta Test Forum

Once you have gained access to the forum, you can:

1. Upload problem reports to a specific library.
2. Download files from the libraries (fixes, updates, etc.).
3. Engage in discussion in the forum messages sections.

Uploading Problem Reports

Once you have found a problem and created a complete problem report using REPORT.EXE, you can upload the file to one of the libraries. A Microsoft Technical Support member will download these reports and forward them to development as needed. Detailed instructions on uploading a file using the Windows Terminal program are included in a later section.

NOTE: Due to the large volume of problem reports that are uploaded, we will be unable to acknowledge receipt of you report at all times.

Downloading Files

You can also download files, updates, etc. from the forum libraries to your computer. Downloading a file is similar to uploading problem reports. Instructions for uploading are discussed in a later section.

Posting Messages and Engaging in Discussion in the Forum

NOTE: While some Microsoft Technical Support Specialists will be in the forum, it is their primary objective to obtain problem reports. You should expect minimal assistance from the Technical Support Specialists during this final phase of the beta test.

Please do not post problem reports as Forum messages. Problem reports should only be uploaded to a library after creating them using REPORT.EXE

The Windows 3.1 beta test area is divided into fifteen different sections where on-going discussions and conversations are carried out among beta testers. Many of the operational problems and questions you might have may be covered in forum discussions with other beta testers. The forum is broken into the following topics:

1. Shell	Discussion of the Windows 3.1 shell and its components, such as File Manager, Program Manager, etc.
2. Setup	Discussion of the setup program and other installation issues.
3. Printing	Comments and issues regarding printer drivers and compatibility with Windows 3.1.
4. Networks	Discussion of network compatibility and configuration issues.
5. Fixes & Updates	Reports from Microsoft of recent fixes, patches and work-arounds.
6. Error Messages	Discussion of UAEs and other errors.
7. Fonts	Compatibility issues and concerns about True Type, ATM and other type managers.
8. Memory Management	Discussion of QEMM, 386MAX and other memory managers and their operation under Windows 3.1.
9. Known Bugs	Reports from Microsoft of recent problems and incompatibilities.
10. Windows Applications	Discussion of various Windows applications and their operation under Windows 3.1.
11. MS-DOS Applications	Discussion of various MS-DOS applications and their operation under Windows 3.1.
12. Other Comments	Suggestions and issues on features, usability of Windows 3.1, and general comments.
13. Problem Form	The Windows 3.1 Problem Report Form is available here in ASCII format for you to download.
14. Dr. Watson Uploads	Upload non-reproducible (only) Dr. Watson logs
15. Message Logs	A record of forum messages by month.

UPLOADING FILES TO COMPUSERVE

The following screen dumps were recorded using the Windows Terminal program and the Xmodem (default Terminal) protocol. Please follow these steps to upload a file to the WINBTU library.

1) Logon to Compuserve.
2) Type !go winbtu
3) To upload a file, enter "LIB" after entering the WINBTU forum:

Note: Do not upload bug reports into the messages section. Upload into the libraries only.

```
MS Windows User BT Forum Menu
 1 INSTRUCTIONS
 2 MESSAGES
 3 LIBRARIES (Files)
 4 CONFERENCING (0 participating)
 5 ANNOUNCEMENTS from sysop
 6 MEMBER directory
 7 OPTIONS for this forum

Enter choice !lib
```

4) Once you are in the library area, you will be prompted for the library section you would like to enter. For example, Enter "10"to access the "Windows Apps" section of the library.

```
MS Windows User BT Forum Libraries Menu
 1 Shell
 2 Setup
 3 Printing
 4 Networks
 5 Fixes & Updates
 6 Error Messages
 7 Fonts
 8 Memory Management
 9 Known Bugs
10 Windows Apps
11 MS-DOS Apps
12 Other Comments
13 Bug Report Form
14 Dr. Watson Uploads
15 Message Files

Enter choice !10
```

If you are an advanced user, you may instead choose the library whose title most closely matches that of your problem. Be sure that you are familiar with the purpose of Library 14, as it isn't a "bug library", but a place for non-reproducible drwatson.log files. You will not be allowed to upload into library 15.

5) Enter "UPL" to begin the upload procedure.

```
MS Windows User BT Forum Library 1
Shell
 1 BROWSE Files
 2 DIRECTORY of Files
 3 UPLOAD a File (FREE)
 4 DOWNLOAD a file to your Computer
 5 LIBRARIES
 6 PREVIEW library
```

```
Enter choice !upl
Suspending connect charges...

Under our Agreement and Operating Rules,
you must own or have sufficient rights to
any information you place on the Service.
```

6. Compuserve will ask you for a filename, and you should type in a filename. This is not really important, except that if it responds to your filename by saying, "This file already exists", please answer no to the next prompt (which will be asking you if you want to overwrite an already existing file), and instead attempt uploading using a different filename:

```
File name: lisxt.txt
```

This person has decided to call their upload "lisxt.txt".,

7. Compuserve will then ask you which Protocol you're going to use. Windows' Terminal defaults to Xmodem, so this would be your selection in this case. Check your Communications software documentation for further details. Your protocol of choice may also be set by entering GO TERMINAL at a CompuServe prompt before your upload session (prior to entering the WINBTU forum). In this example, the user set their transfer mode to XModem from the GO TERMINAL area, prior to entering the WINBTU forum, and thus was not prompted for their preferred transfer protocol.

8. Compuserve will then ask you what kind of file it is (ASCII, Binary, etc.). Choosing binary will work fine, even if the file you're uploading is in report.exe's native text format (meaning it hasn't been "zipped").

```
Transfer types available -
 1 ASCII
 2 Binary
 3 Image
 4 Graphic:RLE
 5 Graphic:NAPLPS
 6 Graphic:GIF

 Enter choice !2
Starting XMODEM receive.

Please initiate XMODEM send
and press <CR> when the transfer
is complete.
```

9. At this point, if you are using Terminal, CompuServe will tell you it is "Starting XMODEM receive" You will then have to go to the Transfers.Send_Binary_File Menu (ALT+T then F) in Terminal and select the file you wish to upload.

10. When you are finished, CompuServe will ask you for a file description. This isn't important, but something must be entered, and concluded with an "/exit" on a line by itself.

```
Enter File Description (up to 464 chars, or about Six 80-character
lines)
Type "/EXIT" when finished:

This is Lisxt.txt
/exit
```

You will be asked for some keywords. Again, these aren't important:

```
Enter Keywords ( or type '?' for HELP ):
none
```

You will be asked for the file's Title:

```
Enter Title (up to 49 chars, <CR> when finished):

Title: Lizst.txt

Then you will be asked to confirm all of this information:
  Title    : Lizst.txt
  Keywords: NONE
  This is Lizt.txt
Is this okay (Y or N)? Y

Resuming connect charges.
```

11. After completing steps 1-10, your file upload is finished. You will then have the choice of uploading more files, or you can press <Enter> consecutively to get back out to the forum. You may also enter "off" from any CompuServe prompt, if you wish to logoff. If you are using Terminal and you've logged off, you will be prompted once again with a HOST: prompt. Simply enter "off" once again from here, and you will complete the logoff procedure.

Software to Help You Use CompuServe

Several third-party software packages are currently available to help facilitate your use of CompuServe. Two of these, TAPCIS and CIM, are in use at Microsoft. These programs can be used exclusively or in conjunction with each other or other programs. Both TAPCIS and CIM operate on MS-DOS based computers.

TAPCIS completely automates CompuServe mail, libraries, and forum messages. It carries out all on-line actions, freeing you to do other things. Once installed, TAPCIS connects you to CompuServe, logs you in, captures information in forums and messages, and then logs you off, allowing you to read off-line.

CIM complements TAPCIS with a menu-driven interface, which allows users to easily read and reply on-line.

For More Information

For sales or other information about CIM/TAPCIS, type GO TAPCIS or GO CIM at any CompuServe ! prompt. Or, call The Support Group, Inc. at 800-872-4768 for additional information on TAPCIS.

APPENDIX B: DR. WATSON UTILITY

DrWatson is a diagnostic tool designed to provide developers with detailed information on the internal state of Windows when a system error such as an application error occurs. DrWatson is installed automatically in all beta releases in the Startup Group by the Windows 3.1 Setup program.

If an application error occurs, DrWatson will automatically create a special file in your WINDOWS directory named DrWATSON.LOG. It will also prompt you for details regarding the circumstances of how the application error occurred. Occasionally Dr. Watson detects a fault that may not be fatal. You will be given the opportunity to "Ignore" the fault, or "Close" the application. When you Ignore a fault, Windows continues without performing the faulting instruction. We provide this opportunity because you may be able to save your work into a NEW file. Do not continue using the application. We recommend you exit Windows at that time, and restart Windows.

If you ignore a fault, you may be presented with the same dialog again. If pressing Ignore a few times doesn't allow continued operation, pressing Close to terminate that application is the only option available.

When you press Close, you will see the normal General Protection Fault box, with an explanation and some numbers. This information is useful in debugging the failure, and it should be sent to Microsoft.

Please send the Windows Beta Support your DRWATSON.LOG file whenever you receive an application error. If it is a reproducible problem, a full problem report should be submitted. Follow the instructions in the following sections to submit the log file to Microsoft. If it is an unreproducible fault, there are three methods of submitting them to Microsoft:

1. Send the Dr. Watson file(s) on a floppy disk to:

   Attn: Dr. Watson Program
   Building 3/2
   ONE MICROSOFT WAY
   REDMOND, WA 98052-9953

2. Send the files via INTERNET to our in-house e-mail alias, "watson".
3. Upload these files into library 14 on the Windows 3.1 Beta Forum.

Please do not fax these files to Microsoft.
It is preferred that you submit this information in electronic form since it is easier to process. You should delete the file DRWATSON.LOG after sending the information to Microsoft, because Dr Watson will append any new data to this same file.

MS-PCA 2559198

CONFIDENTIAL

Appendix C: Commonly Asked Questions

Q. What if I have a bad disk?
A. Call 206-936-7154

Q. What about contacting third parties?

Q. What do I need to do to get a free copy of Windows 3.1 once it is released to the public?

Q. What about my NDA?

Q. What about updated builds?